ariadne

the System i specialists

Slipstream

User Guide

Version V5R2M0
April 2009

ariadne software ltd, cheltenham, england

Table of Contents

Introduction 6

Applications 8

Automated distribution of reports and documents 8

Integration with your normal batch jobs 8

Data Sharing 9

File Formats 9

Excel Format 10

XML Format 10

Delimited ASCII Text 10

HTML (Hypertext Mark-up Language) 10

Fixed ASCII Text 10

PDF 10

What's new in V5R2M0? 11

System Requirements 12

Upgrade Notes 14

Why V5R2M0? 14

License Keys 15

Warning/Disclaimer 15

Minimum OS/400 Release Level 16

Product Library 16

Licensed Program Id 16

Modifications in V5R2M0 requiring code changes 16

Installation 18

Method 1: Installing Slipstream using FTP 18

Method 2: Installing Slipstream using the IFS 20

Maintenance 23

Method 1: Installing a PTF using FTP 23

Method 2: Installing a PTF using the IFS 25

Where Did My Output Go? 28

The TOSTMF parameter 28

Understanding IFS path names 28

Choosing where to store your output 31

Root File System 32

QDLS File System 32
QNTC File System 33
Typical Solutions 34
Slipstream Variables 36
Slipstream Functions 37
Excel Placeholders 42
Using Slipstream 44
Getting Started 44
Using styles 44
Using conditional formatting 45
Using encrypted passwords 45
CVTDBFxxxx Command Parameters 47
FROMFILE – From database file 47
TOSTMF – To stream file 50
TOFMT – To format 52
STMFOPT – Stream file option 54
RPLXLSSHT – Replace Excel worksheet names 56
FORMAT – Format specification 57
FROMRCD – From record number 58
TORCD – To record number 59
SEPCHAR – Separator character 60
SQL – SQL statement options 61
SQLSRC – SQL source options 62
QRYDFN – Query/400 options 64
QRYFILE – Query file 68
QMQRY – QM Query options 69
FTP – FTP parameters 71
EMAIL – Email the output? 75
EMAILOPT – Email options 76
EMAILFROM – Email sender information 80
EMAILTO – Email recipient(s) 82
EMAILMSG – Email message 84
RCDFMT – Record format 86
INCLFLD – Include fields 87
EXCLFLD – Exclude fields 88
EXCEL – Excel options 89

XLSPRPRTY – Document properties 96
XLSPROTECT – Excel worksheet protection 98
XLSPRINT – Excel print setup 100
XLSADJUST – Adjust pages to 103
XLSFIT – Fit pages to 103
CSV – CSV options 105
FIXED – Fixed text options 109
HTML – HTML options 113
XML – XML options 117
XMLNAMESPC – XML namespace options 120
XMLSCHEMA – XML schema options 121
XMLSTYLING – XML styling options 123
CSSSTYLING – CSS stylesheet options 125
HEADER – Header row 127
DFNSTYLES – Define styles 130
CNDFMTGRP – Conditional formatting groups 148
CNDFMTRULE – Conditional formatting rules 150
FLDSTYLES – Field styles and attributes 158
SORT – Sort specifications 164
QRYSLT - Query selection expression 165
DBFCCSID – Database file CCSID 166
STMFCODPAG – Stream file code page 167
UNICODE 169
DBFDATFMT – Database date format 170
AUT 172
INHERITAUT 173
CVTXLSDBF Command 174
FROMSTMF – Excel file to convert 174
FROMSHEETS –Worksheet(s) to convert 174
TOFILE – File to receive output 174
TOMBR – Member to receive output 175
TOCCSID – CCSID to convert to 175
BLANKS – Output blank cells 175
ROUNDING – Floating point rounding 176
CVTXLSDBF output format 176
DSPENCPWD Command 179

PWD– Password to encrypt 179
ZIPDTA Command.....180
FROMFILE –Files to zip 180
TOZIP – Zip file name 180
STMFOPT – Stream file option 180
CPRLVL – The data compression level..... 180
PWD – Password 181
UNZIPDTA Command 182
FROMZIP –From zip file..... 182
FROMFILE –Files to unzip 182
TODIR – Unzip to directory 182
TOFILE – Unzip to file 182
REPLACE – Replace existing files..... 182
PWD – Password 182
STMFCODPAG – Stream file code page 183
AUT – Public data authority..... 183

Introduction

Slipstream helps IBM system i sites deliver high-quality information to their customers and users quicker and more simply than was possible previously.

Slipstream does this by providing tools for exporting system i database tables as PC-format stream files. As of V5R2M0, it also provides a way of extracting information from Excel-format spreadsheets into system i database format.

Slipstream is now available in two forms: as a standalone licensed program; and as a module in the ***CoolSpools Plus*** bundle. ***CoolSpools Plu***s bundles ***CoolSpools***, ***Communiqué***, ***Slipstream*** and ***CoolTools*** as a single licensed program using a single product library. If you are installing CoolSpools Plus (licensed program 5COOLSP), refer to the User Guide for CoolSpools Plus for details of how to install ***CoolSpools Plus***.

The system i database (UDB for system i) is an advanced, function-rich, easily managed relational database allowing large quantities of data to be stored securely and retrieved and updated rapidly and efficiently. However, today, many of our customers and users want to be able to access and manipulate the data held in that database using tools such as spreadsheets, PC databases and Business Intelligence tools. Enabling their customers and users to access data held on the system i in the form in which they need it is a challenge that many system i and system i sites face on daily basis.

One frequently used approach to this problem is to allow users to import data from the system i database into PC application using tools such as iSeries Access file transfers and ODBC queries.

However, these solutions have a number of drawbacks, mainly related to the fact that they are “pull” technologies. In other words, an iSeries Access file transfer or an ODBC query has to be initiated from the PC end and cannot easily be driven from the system i. This has a number of unfortunate consequences:

- Your users have to be in the office to press the buttons. This means that these types of operations typically have to be run during the day. This imposes an unnecessary load on your system i and your network when they are at their busiest. It also means that your users waste time preparing data for use. Wouldn’t it be better if the data was ready waiting for them when they arrive in the office?
- Your users have to push the buttons at the right time. What happens if the job that creates or updates the data they want hasn’t finished running yet? Do they get yesterday’s data instead? Or might data get lost, corrupted or duplicated? Wouldn’t it be better if you could integrate and schedule the preparation of PC data with your batch processes so that the PC data isn’t created until the system i is ready?
- Your users have to make sure to push the right buttons! Running regular file transfer and queries can be a repetitive, tedious, error-prone chore. Human beings are fallible and mistakes will inevitably occur from time to time. Wouldn’t it be better if these processes were automated?

Slipstream by contrast is a “push” technology. It addresses these problems by providing a set of integrated tools to enable PC-format data to be extracted from the system i database entirely on the system i, without the need for a PC or any user intervention. This means that the entire process can be fully automated and controlled programmatically, and can be easily integrated with your existing applications and batch schedules.

Slipstream can process any of the following as input sources:

- system i physical files
- system i logical files
- system i DDM files
- ad hoc SQL query statements
- SQL query statements stored in a source file member
- Query/400 queries
- Query Management (QM/400) queries

Slipstream can output the results or contents of these input sources in the form of stream files in any of the following file formats:

- **Excel**. As of V5R2M0, the flavors of Excel supported are:
  - BIFF 8 (Excel 97-2003 .xls format)
  - Office Open XML (Excel 2007 .xlsx) ***New in V5R2M0!***
- **XML.** ***New in V5R2M0!***
- **HTML**
- **Delimited ASCII text**, including:
  - Comma Separated Variable (CSV)
  - Tab Separated Variable (TSV)
  - other delimited text formats (e.g. semicolon)
- **Fixed-width text**

This means that you can now execute a piece of SQL and save the output directly as an Excel spreadsheet, simply by running one command on your system i!

Similarly, you could run a query and save the output as an ASCII CSV file, ready to be loaded into Lotus 123. Or if you want to display the latest sales figures on your Intranet, rather than printing a paper report as you might have done before, saved the output from you sales reporting program as a temporary database file, and use ***Slipstream*** to convert it to HTML format.

Stream files created by ***Slipstream*** can be saved anywhere in your system i's IFS (Integrated File System). This means they can either be held on the system i itself (using the root or QDLS file systems), or transferred seamlessly to a Windows server (using the QNTC file system) or to another system i (use the QFileSvr.400 file system). See the section entitled "Using the IFS" for details of how to transfer data directly to an NT server when running ***Slipstream.*** ***Slipstream*** can also send its output to another system using FTP.

When processing an existing system i physical or logical file as input, ***Slipstream*** provides a number of additional functions which make delivering high-quality, easily accessible data to your users even easier:

- Record sorting – records can be sorted by multiple key fields in ascending or descending sequence

- Record selection – records can be selected or omitted based on criteria that you define in a familiar format
- Field inclusion/exclusion – you can include all fields in the file being processed, specify a list of fields to include or a list of fields to exclude
- Field sorting – you can specify the order in which fields are to be presented by listing them in the order you want them
- Formatting – date, time and number formats can be managed and edit codes and edit words are reflected in Excel formatting.
- User-defined named styles allowing you to specify formatting of Excel columns and HTML table data such as colors, fonts etc. ***New in V5R2M0!***

Applications

Slipstream revolves a series of related commands which can take any system i database file (i.e. DB2 UDB for system i file or table) and convert it to formats that can be used by popular PC applications such as spreadsheets, databases and browsers.

Here are a few possible ways ***Slipstream*** can help you deliver data to your users more efficiently and reliably.

Automated distribution of reports and documents

Still sending reports to your users on paper? Still distributing data to customers via snail mail?

Wouldn't your users rather have the same information in electronic form so they can load it into their PC spreadsheets without having to re-key it? ***Slipstream*** can help you modify your applications so you can deliver data to your users in the form they need. All that is needed is that you change your programs to output to a database file rather than to a spooled file. You can then use ***Slipstream*** to create an Excel spreadsheet, CSV or HTML or fixed text file from your database file

Files created by ***Slipstream*** can be e-mailed to users, colleagues and customers, rather than sent out on paper, making their lives easier whilst saving you money and simplifying your business processes.

Integration with your normal batch jobs

Imagine you have Excel power-users in your finance department who need to analyze your company's sales figures from the previous week each Monday morning. To get the information from the sales database into Excel, your users could perform an iSeries Access file transfer or run an ODBC query. "Isn't that the easiest way?" you might ask. However, ***Slipstream*** offers some significant advantages over these methods:

- Data is ready for your users as soon as they want it. Run your conversions overnight and in the morning your users simply need to connect to the system i and open a file from the IFS or a PC server. They don't need to waste time running file transfers or ODBC queries.
- Data can be prepared when convenient to you. Conversions can be easily integrated with your normal overnight or weekend batch jobs, and therefore run out of hours when the load on your system and network is lowest. There's no need for a PC at all and no need for complicated, error-prone system i-to-PC communications to trigger file transfers or ODBC queries.

- Conversions don't take place until the data is ready. Run the conversion step immediately after the step that creates the data to ensure that the two occur in the right sequence.
- Since the conversion parameters are built into your applications, you can be sure that the conversion will be done correctly every time, automatically. You don't have to rely on your users to run the transfer or the query correctly each day or each week, and they have to remember what buttons to press and which selections to make each time.

For example, rather than having your users run ODBC queries against the sales database on a Monday morning, you could run a job over the weekend to extract information about sales made the previous week. This could be as simple as running a Query/400 query. With ***Slipstream*** you can run the query and save the output as an Excel, CSV file, HTML or ASCII text file in your IFS. Your users only need a network drive assigned to the right system i and the necessary permissions to access the data. They can simply load their spreadsheet application on Monday morning, open the file previously created using Slipstream, and away they go.

The beauty of this approach is that the system i controls, schedules, and manages the creation of the PC-format data entirely. It saves your users time and effort, saves you worries about whether the users will get their file transfer right or run their ODBC query at the wrong time, and saves the system i from unnecessary workload during office hours.

Data Sharing

Often, many of your users will want the same data. This can often mean that different users will be running the same or very similar queries or file transfers against your database all in the same day. This is very inefficient. Why not run that query for them once, overnight, and save the results as an Excel, CSV or HTML file to a central server so the information can be shared by everyone?

Alternatively, you could publish these files on your Web site for customers worldwide to see, or on a secure Intranet or Extranet.

Stream files created by ***Slipstream*** can be saved anywhere in your system i's IFS (Integrated File System).

File Formats

Let's have a closer look at how ***Slipstream*** achieves all this.

Slipstream does this through a set of simple commands such as CVTDBFXLS (Convert Database File to Excel) and CVTDBFXML (Convert Database File to XML). These commands take a system i database file (physical or logical) and convert it to a stream file in a particular PC file format (e.g. Excel for CVTDBFXLS and XML for CVTDBFXML). The stream file can be saved anywhere in the IFS (Integrated File System). See the section entitled "Using the IFS" below for details.

Please note that it is not necessary for the stream file to be stored either temporarily or permanent on your system i disk: the IFS provides the functionality to write the stream file directly to an NT or UNIX server, if you prefer to store your files there for your users to access. Alternatively, store the stream files locally in the root file system of your system i, and anyone with the right authorities and a network drive attached can read them

You can choose several different formats for your files, depending on your particular requirements.

Excel Format

If you use the ***CVTDBFXLS*** command, you can create an Excel (™) spreadsheet from your database file. The default is to create a BIFF 8 file (compatible with Excel 97, Excel 2000, Excel XP and above). Alternatively, you can choose to create an Excel 2007 .xlsx (Office Open XML) file.

Files created by ***CVTDBFXLS*** in Excel format can be opened directly by any application with supports the Excel file format, including Microsoft Excel but many others besides, such as Lotus 123 and MS Works spreadsheet.

XML Format

The ***CVTDBFXML*** command converts database files to XML format. You can select, for any field in the database file, whether it should be converted to an attribute or an element in the XML file.

Delimited ASCII Text

The ***CVTDBFCSV*** command creates delimited ASCII text files, typically, but not limited to, CSV (comma-separated variable file). This format is ideal for loading reports containing columns of numbers into a spreadsheet, Business Intelligence tool or other application for further manipulation. ***CVTDBFCSV*** can use any field delimiter you like (by default a comma, but also tabs, semicolons, blanks etc.) and any string separator you specify (by default a double quote “), allowing you to generate files in the precise format required by your PC application or national environment.

HTML (Hypertext Mark-up Language)

If your want to view your data in a browser, such as Microsoft ® Internet Explorer or Mozilla Firefox, ***CVTDBFHTML*** is the command you need.

HTML is the language in which web pages are normally written. ***CVTDBFHTML*** can convert your database files to a formatted HTML table, which provides a neat way of displaying your data on the Internet, or on your company Intranet or Extranet.

Fixed ASCII Text

CVTDBFTXT will converts your system i database files to a simple ASCII text file with fixed-width data columns without delimiters or separators. This format can be readily processed by user-written applications so long as the file layout is known in advance, and can be useful for exchanging data with business partners in EDI operations.

PDF

If you have CoolSpools or CoolSpools PLUS, ***CVTDBFPDF*** will run the query you specify or an auto-generated query against the database file you specify and convert the output to a PDF file.

What's new in V5R2M0?

Highlights include:

- **XML**
  - Slipstream can now generate XML by means of the new CVTDBFXML command
  - A simple schema (or XSD or DTD) can be automatically generated or the XML can be linked to an existing schema
  - A simple stylesheet (XSLT or CSS) can be automatically generated or the XML can be linked to an existing stylesheet

- **Office Open XML**
  - Slipstream can now optionally generate Excel 2007 Office Open XML format files (.xlsx)
  - BIFF 8 (Excel 97+) .xls files is still the default Excel format
  - Support for BIFF 5 (Excel 95) files is now withdrawn.

- **New format-specific commands**
  - **CVTDBFXLS, CVTDBFXML**, **CVTDBFHTML**, **CVTDBFTXT** and **CVTDBFCSV** generate Excel, XML, HTML, fixed-width text and delimited text respectively
  - Introducing these new commands has allowed parameter definitions to be reorganized and rationalized to improve ease of use without impacting existing applications.
  - CVTDBFSTMF is still supported but its use is now discouraged. Most new features are available only through the new format-specific commands and CVTDBFSTMF will not be further enhanced.

- **User-defined named styles**
  - When converting to Excel, XML and HTML formats, named styles can be defined on the new DFNSTYLES parameter
  - The styles can be associated with particular types of data (details, headings, titles etc.) or, using the FLDSTYLES parameter, with individual fields to control the visual formatting of the output such as font, text color, background color, numeric formatting etc.

- **Conditional formatting**
  - When converting to Excel, you can apply conditional formatting to selected columns or entire rows.
  - For example, set the color of rows based on the value of a specified field: red for high values, green for low values etc. or make certain fields bold if rules you specify evaluate to true.
  - Currently available only when converting to Open Office XML format (.xlsx)

- **User-defined column headings and widths**

- As well as letting you define field-level styling, the FLDSTYLES parameter lets you specify field-level attributes such as: column headings, column width, date

- **Improved support for dates and times in numeric and character fields**
  - FLDSTYLES also lets you identify character and numeric fields as containing dates and times so that Slipstream knows to convert them accordingly.

- **Slipstream variables and Slipstream functions**
  - Use Slipstream variables and Slipstream functions to generate file names, email addresses etc at run time from database file attributes, job attributes etc.

- **Secure FTP (FTPs)**
  - Support for FTP over SSL when using TOSTMF(*FTP) to send the output to a remote system using FTP

- **Excel export to database format**
  - By means of the CVTXLSDBF (Convert Excel to Database File) command, Slipstream can now extract cell data from an Excel spreadsheet (BIFF8 or Office Open XML) and write that data to a database "outfile".
  - Your applications can then process the outfile to import the data into your database or do other work.

- **Encrypted passwords**
  - Where a password can be specified on a command parameter (e.g. FTP connection, zip file), the password can be supplied as an encrypted hex string to avoid the need to hold passwords in plain text form in source code
  - DSPENCPWD (Display Encrypted Password) command generates the encrypted form of a password to be used

- **ZIPDTA and UNZIPDTA commands**
  - In order to help you with your distribution of files created by Slipstream, Slipstream now includes a simple zip facility in the form of the ZIPDTA command with a corresponding unzip facility (UNZIPDTA command)
  - This facility uses the industry-standard ZLIB library to produce zip format files compatible with WinZip and other zip utilities
  - These allow you to compress stream files, save files and database file members
  - Optionally, zip files can be encrypted/password-protected

System Requirements

- A system i or running IBM i (OS/400) V5R2M0 or above.
- 20 Mb of system i disk space.

- **No** PC is required.

Upgrade Notes

Please read the following notes carefully before upgrading to Version 4 from an earlier version of Slipstream.

Why V5R2M0?

The previous version of Slipstream had a licensed program id of 4STMEXP and a version number of V4R1M0. The new version of Slipstream has a licensed program id of 5SLPSTM and a version number of V5R2M0. While at first sight this may seem strange, there is method to this madness!

As far as the version number is concerned, the intention is to try to eliminate the confusion that has arisen in the past in relation to the version number of the modules that make up the CoolSpools PLUS bundle. Each of the various modules (CoolSpools, Slipstream, Communiqué and CoolTools) that make up CoolSpools PLUS V5R2M0 is now likewise designated V5R2M0. By a happy coincidence, the minimum OS/400 release required to run these products is also V5R2M0.

In relation to the licensed program id, the previous id (4STMEXP) was short for “Slipstream Exporter”, reflecting the fact that Slipstream could only export data from the iSeries database to other formats. The new id (5SLPSTM) reflects the fact that Slipstream can now do more than that and includes the facility to import data into an iSeries database table from Excel.

If you have installed the **CoolSpools Plus** product, which bundles Slipstream with a number of other modules, the product library will be either COOLSPV5R1 (CoolSpools PLUS V5R1) or COOLSPV5R2 (CoolSpools PLUS V5R2). CoolSpools PLUS V5R1 incorporates Version 4 of Slipstream, while CoolSpools PLUS V5R2 incorporates Slipstream V5R2.

If your command is in a library other than those shown, you are either running an unsupported, unlicensed version or have moved or copied the command object from its original location. Contact support@ariadnesoftware.co.uk for assistance.

To determine which version of Slipstream you are running, check the name of the library in which the CVTDBFSTMF command object you are using resides, e.g.:

DSPOBJD OBJ(CVTDBFSTMF) OBJTYPE(*CMD)

The library name corresponds to the version of Slipstream as shown in the table below:

Library Name	Version
STMEXPV2R1	Slipstream Version 2
STMEXPV3R1	Slipstream Version 3
STMEXPV4R1	Slipstream Version 4
SLPSTMV5R2	Slipstream V5R2M0
COOLSPV5R1	CoolSpools PLUS V5R1M0
COOLSPV5R2	CoolSpools PLUS V5R2M0

If your command is in a library other than those shown, you are either running an unsupported, unlicensed version or have moved or copied the command object from its original location. Contact support@ariadnesoftware.co.uk for assistance.

License Keys

You are entitled to upgrade to V5R2M0 of Slipstream free of charge if:

- the machine on which you wish to run Slipstream V5R2M0 has a valid license for an earlier version of Slipstream

and

- you are either in your first 12 months' maintenance period after purchase or have paid your latest annual maintenance invoice.

If you wish to upgrade, you can simply download the software from www.ariadnesoftware.co.uk and install it according to the instructions contained in the "Installation" section of this User Guide. However, if you licensed an earlier version of Slipstream, you will need to request a license key for the new version. Simply e-mail support@ariadnesoftware.co.uk and ask for your key for V5R2M0 of Slipstream. Please quote your system serial number(s) and processor group code(s) in your e-mail. These are shown at the top of the WRKLICINF screen.

Without a license key, Slipstream V5R2M0 will allow you a 30-day grace period and will then no longer run.

If you have not paid your annual maintenance invoice, and if you need longer than 30-days to test V5, we will, on request, send you a temporary license key to extend the trial period.

If you require additional temporary license keys to assist with testing Slipstream V5, or if you run into any problems during your testing, please do not hesitate to contact us at support@ariadnesoftware.co.uk.

Warning/Disclaimer

We recommend strongly that all production applications are re-tested thoroughly using the new version in your development environment before you switch over to running the new version in your production environment.

All Slipstream versions are packaged as separate licensed programs and install into different libraries. This means that all versions of Slipstream can coexist and run alongside one another on the same machine. You can switch an application from using one version to using another simply by changing the library list of the job to include the appropriate version library or by specifying a different library name when you run the command. Hence it is quite a simple matter to test your applications using the new version while continuing to run the older version in production.

Please note that while ariadne makes every effort to ensure that Slipstream functions in the same way with the same parameters from one version to the next, it is not possible to guarantee this. This is why you should re-test your applications against a new version before going live with it as it is possible that in some cases different parameter settings will be necessary to obtain the same results as before.

ariadne software accepts no responsibility for any damage, expense or loss of income incurred as a result of unforeseen and unwanted effects resulting from installing new versions of its software or applying PTFs.

Minimum OS/400 Release Level

The minimum OS/400 release level required to run V5R2M0 of Slipstream is OS/400 **V5R2M0.**

If you are running V5R2M0 or an earlier version of OS400, you will not be able to install V5R2M0 of Slipstream.

Product Library

The product library for the standalone version of Slipstream V5R2M0 is now **SLPSTMV5R2**.

However, if you have installed the **CoolSpools Plus** product, it is **COOLSPV5R2**. CoolSpools Plus bundles Slipstream with CoolSpools, Communiqué and CoolTools.

You may need to change library lists in job descriptions and other system objects in order to pick up the new version of the code rather than the old.

This change of library name has the advantage that it allows you to run both V5R2M0 and the earlier versions on the same machine. You are therefore able to test V5R2M0 before swapping your production applications over to the new version, as we strongly advise you to do.

Licensed Program Id

The licensed program identifiers for the various standalone versions of Slipstream and the versions of CoolSpools PLUS that include Slipstream are as follows:

Licensed Program Id	Version
2STMEXP	Slipstream Version 2
3STMEXP	Slipstream Version 3
4STMEXP	Slipstream Version 4
5SLPSTM	Slipstream V5R2M0
5COOLSP	CoolSpools PLUS V5R1M0 or V5R2M0

Modifications in V5R2M0 requiring code changes

This section lists only those changes that may require existing users to modify parameters or re-code applications that run Slipstream. Refer to the “What’s New” section for new features and enhancements.

The CVTDBFSTMF command is little changed in this release and it is not anticipated that any code that references CVTDBFSTMF will need to be change to run under V5R2M0. However, you are encouraged to switch over to running the format-specific commands instead of CVTDBFSTMF at the first opportunity. In addition to changing the command name and removing the obsolete TOFMT parameter, you are highly likely to have to make other changes as there have been numerous small changes to parameter values and defaults. Refer to the documentation of the command parameters below for details.

Installation

If you are installing CoolSpools Plus, please refer to the installation instructions in the CoolSpools Plus User Guide. These instructions apply only where you are installing the standalone version of ***Slipstream***.

If you downloaded ***Slipstream*** from the Internet, or received the product via e-mail, you will have received a zip file called **SLPSTMV5R2.zip**. This zip file contains everything you need to get started with ***Slipstream***. This section describes two possible procedures for installing ***Slipstream*** on your system i. We are hoping to simplify and automate this area in the near future. However, for the moment, please select one of the two methods described below and follow the steps carefully and in the order indicated. Steps to be carried out on the PC are shown in **green**, steps to be carried out on the system i in **blue**.

Method 1: Installing Slipstream using FTP

This method uses FTP (File Transfer Protocol) to transfer the Slipstream product files from your PC to your system i. It is assumed that you have stored the SLPSTMV5R2.zip file which you downloaded from the Internet or received via e-mail on your PC and that you have enabled an FTP link between your PC and your system i.

Step 1

On your PC, create an install directory for Slipstream, either within Windows Explorer, or from the DOS prompt, e.g.:

```
md c:\SLPSTMV5R2
```

Save or copy the zip file that you downloaded or received into this directory.

Step 2

On your PC, unzip the contents of the zip file into the directory you just created using pkunzip, WinZip, Norton UnZip etc.

Step 3

On your system i, decide which library you wish to use for the install procedure and create a save file in that library, e.g.

```
crtsavf lib_name/SLPSTMV5R2
```

where **lib_name** is the name of the library you chose to use.

Step 4

On your PC, start an FTP session to your system i. You may use FTP packages such as CuteFTP or simply go to the DOS prompt and type:

```
ftp AS400_name
```

where AS400_name is the name of your system i.

You will be prompted to enter your user id and password. Please ensure that you sign on with a user id that has adequate authority. We recommend that you sign on as the Security Officer QSECOFR.

Issue the following FTP commands within the FTP session:

```
binary
```

This command instructs FTP not to carry out any conversion on the data to be sent to the system i.

quote site na 0

This makes sure the system i is using naming format 0 (library naming). This is important if your FTP attributes (CHGFTPA command NAMEFMT parameter) have the initial naming format set to *PATH (IFS path naming) rather than *LIB (library/file.member naming).

cd lib_name

where lib_name is the name of the library in which you created the save file in Step 3. This command makes lib_name your current library on the system i.

lcd dir_name

where dir_name is the name of the directory on your PC in which you unzipped the install files earlier. This command makes dir_name your current directory on the PC.

put SLPSTMV5R2.sav SLPSTMV5R2

This command transmits the contents of the file SLPSTMV5R2.sav, which was earlier unzipped from SLPSTMV5R2.zip, and stores them in the save file called SLPSTMV5R2 which you created earlier on your system i.

Step 5

On your system i, ensure that the previous step worked by displaying the contents of save file SLPSTMV5R2:

dspsavf lib_name/SLPSTMV5R2

If you see the message **(No records in save file)** then something has gone wrong during the preceding steps. Try following them again. If you still cannot get this procedure to work, contact us at **support@ariadnesoftware.co.uk.**

Step 6

On your system i, issue the following command to restore the Slipstream licensed program to your system:

```
rstlicpgm      licpgm(5SLPSTM)
               dev(*savf)
               option(*base)
               rstobj(*all)
               lng(*primary)
               output(*none)
               rls(*first)
               replacerls(*only)
               savf(lib_name/SLPSTMV5R2)
```

where lib_name is the name of the library in which you created the save file in Step 3.

Step 7

If you have downloaded Slipstream from the Internet, you can use the product without a license key for up to 30 days. At the end of that period, Slipstream will notify you that it can no longer be used without a license key.

To obtain a license key contact ariadne at **sales@ariadnesoftware.co.uk.**

If you have been issued with a license key, proceed as follows.

On your system i, issue the following command to add a license key to enable you to use the Slipstream licensed program:

```
addlickey      lickeyinp(*prompt)
               prdid(5SLPSTM)
               licterm(V5R2)
               feature(5001)
               serial(*local)
               prcgrp(Ppp)
               lickey(XXXXXX YYYYYY ZZZZZZ)
               usglmt(*nomax)
               expdate(exp_date)
               vnddta(*none)
```

where: **XXXXXX YYYYYY ZZZZZZ** is the license key you received by e-mail from us
exp_date is the expiry date mentioned in the e-mail
and: **Ppp** is the processor group mentioned in the e-mail

Slipstream is now ready for you to use!

If you have any trouble at any point in this procedure, please contact us by e-mail at **support@ariadnesoftware.co.uk.**

Method 2: Installing Slipstream using the IFS

This method uses a network drive assigned to the IFS of your system i to transfer the Slipstream product files from your PC to your system i. It is assumed that you have stored the SLPSTMV5R2.zip file (which your downloaded from the Internet or received via e-mail) on your PC.

Step 1

On your PC, create an install directory for Slipstream, either within Windows Explorer, or from the DOS prompt, e.g.:

```
md c:\PC_dir
```

where PC_dir is the name of the directory you wish to use for this installation.

Save or move the zip file that you downloaded or received into this directory.

Step 2

On your PC, unzip the contents of the zip file into the directory you just created using pkunzip, WinZip, Norton UnZip etc.

Step 3

On your PC, ensure that you have a network drive assigned to a directory in the root file system of your system i' IFS. This can be done in Windows Explorer or (My) Computer by selecting Tools, Map Network Drive.

Create a directory in the IFS to receive the Slipstream files. This can be done in Windows Explorer by selecting File, New, Folder, or at the DOS prompt as follows:

```
md i:\IFS_dir
```

where **I:** is the network drive assigned to your system i and IFS_dir is the name of the directory you wish to use for this installation.

Step 4

Copy the contents of your PC's install directory created in Step 1 above to the IFS directory created in Step 3 above using Windows Explorer, or from the DOS prompt as follows:

```
copy c:\PC_dir\*.* I:\IFS_dir
```

where **PC_dir** is the name of the directory on your PC, **I:** is the network drive assigned to your system i and **IFS_dir** is the name of the directory you wish to use for this installation.

Step 5

On your system i, decide which library you wish to use for the install procedure and create a save file in that library, e.g.

```
crtsavf lib_name/SLPSTMV5R2
```

where **lib_name** is the name of the library you chose to use.

Step 6

On your system i, copy the contents of the SLPSTMV5R2.sav file from the IFS directory to the save file just created, by issuing the following command:

```
cpyfrmstmf      fromstmf('/IFS_dir/SLPSTMV5R2.sav')
                tombr('/qsys.lib/lib_name.lib/SLPSTMV5R2.file')
                mbropt(*replace)
                cvtdta(*none)
                endlinfmt(*fixed)
                tabexpn(*no)
```

where **IFS_dir** is the name of the IFS directory into which you copied the install files in Step 4 above, and **lib_name** is the name of the library in which you created the save file in Step 5 above.

Step 7

On your system i, ensure that the previous step worked by displaying the contents of save file SLPSTMV5R2:

```
dspsavf lib_name/SLPSTMV5R2
```

where lib_name is the name of the library in which you created the save file in Step 5.

If you see the message **(No records in save file)** then something has gone wrong during the preceding steps. Try following them again. If you still cannot get this procedure to work, contact us at **support@ariadnesoftware.co.uk.**

Step 8

On your system i, issue the following command to restore the Slipstream licensed program to your system:

```
rstlicpgm       licpgm(5SLPSTM)
```

dev(*savf)
option(*base)
rstobj(*all)
lng(*primary)
output(*none)
rls(*first)
replacerls(*only)
savf(lib_name/SLPSTMV5R2)

where lib_name is the name of the library in which you created the save file in Step 5.

Step 9

If you have downloaded Slipstream from the Internet, you can use the product without a license key for up to 30 days. At the end of that period
Slipstream will notify you that it can no longer be used without a license key.

To obtain a license key contact ariadne at sales@ariadnesoftware.co.uk.
If you have been issued with a license key, proceed as follows.

On your system i, issue the following command to add a license key to enable you to use the Slipstream licensed program:

addlickey
lickeyinp(*prompt)
prdid(5SLPSTM)
licterm(V5R2)
feature(5001)
serial(*local)
prcgrp(Ppp)
lickey(XXXXXX YYYYYY ZZZZZZ)
usglmt(*nomax)
expdate(exp_date)
vnddta(*none)

where: **XXXXXX YYYYYY ZZZZZZ** is the license key you received by e-mail from us
exp_date is the expiry date mentioned in the e-mail
and: **Ppp** is the processor group mentioned in the e-mail

Slipstream is now ready for you to use!

If you have any trouble at any point in this procedure, please contact us by e-mail at **support@ariadnesoftware.co.uk.**

Maintenance

Periodically we will release Program Temporary Fixes (PTFs) in order to add new features or to resolve any problems found with ***Slipstream***.

If you have installed CoolSpools Plus, please refer to the installation instructions in the CoolSpools Plus User Guide. These instructions apply only where you have installed the standalone version of ***Slipstream***.

PTFs are supplied in the form of a system i save file stored within a zip file, and can be downloaded from the Download section of the ariadne web site at **http://www.ariadnesoftware.co.uk.**

The procedure for loading and applying a PTF for Slipstream is as follows.

First download the PTF you require from the Internet. The PTF will be held in a file called Q5SLnnnn.zip, where nnnn is the PTF id.

This section describes two possible procedures for copying the system i save file held in this zip file to your system i. We are hoping to simplify and automate this area in the near future. However, for the moment, please select one of the two methods described below and follow the steps carefully and in the order indicated. Steps to be carried out on the PC are shown in **green**, steps to be carried out on the system i in **blue**.

Method 1: Installing a PTF using FTP

This method uses FTP (File Transfer Protocol) to transfer the PTF file from your PC to your system i. It is assumed that you have stored the zip file which you downloaded from the Internet on your PC and that you have enabled an FTP link between your PC and your system i.

Step 1

On your PC, create a directory for the PTF installation, either within Windows Explorer, or from the DOS prompt, or re-use the directory you used to install originally, e.g.:

```
md c:\SLPSTMV5R2
```

Save or copy the zip file that you downloaded or received into this directory.

Step 2

On your PC, unzip the contents of the zip file into the directory you just created using pkunzip, WinZip, Norton UnZip etc.

Step 3

On your system i, first sign off then sign back on again to ensure that you do to have any Slipstream objects in your QTEMP library. Then decide which library you wish to use for the PTF install procedure and create a save file in that library. The name of this save file should be Q5SLnnnn, where nnnn is the PTF id, e.g.

```
crtsavf lib_name/Q5SL0001
```

where **lib_name** is the name of the library you chose to use.

Step 4

On your PC, start an FTP session to your system i. You may use FTP packages such as CuteFTP or simply go to the DOS prompt and type:

ftp AS400_name

where AS400_name is the name of your system i.

You will be prompted to enter your user id and password. Please ensure that you sign on with a user id that has adequate authority. We recommend that you sign on as the Security Officer QSECOFR.

Issue the following FTP commands within the FTP session:

binary

This command instructs FTP not to carry out any conversion on the data to be sent to the system i.

quote site na 0

This makes sure the system i is using naming format 0 (library naming). This is important if your FTP attributes (CHGFTPA command NAMEFMT parameter) have the initial naming format set to *PATH (IFS path naming) rather than *LIB (library/file.member naming).

cd lib_name

where lib_name is the name of the library in which you created the save file in Step 3. This command makes lib_name your current library on the system i.

lcd dir_name

where dir_name is the name of the directory on your PC in which you unzipped the install files earlier. This command makes dir_name your current directory on the PC.

put Q5SLnnnn.sav Q5SLnnnn

This command transmits the contents of the file Q5SL0001.sav, which was earlier unzipped from Q5SL0001.zip, and stores them in the save file called Q5SLnnnn which you created earlier on your system i.

Step 5

On your system i, ensure that the previous step worked by displaying the contents of save file Q5SLnnnn:

dspsavf lib_name/Q5SLnnnn

If you see the message **(No records in save file)** then something has gone wrong during the preceding steps. Try following them again. If you still cannot get this procedure to work, contact us at **support@ariadnesoftware.co.uk.**

Step 6

On your system i, issue the following command to load the PTF on to your system:

```
lodptf       licpgm(5SLPSTM)
             dev(*savf)
             select(5SLnnnn)
             savf(lib_name/Q5SLnnnn)
```

where lib_name is the name of the library in which you created the save file in Step 3 and nnnn is the PTF id.

Step 7

On your system i, issue the following command to apply the PTF to your system:

```
apyptf      licpgm(5SLPSTM)
            select(5SLnnnn)
            apy(*temp)
```

where: **nnnn** is the PTF

If you have any trouble at any point in this procedure, please contact us by e-mail at **support@ariadnesoftware.co.uk.**

Method 2: Installing a PTF using the IFS

This method uses a network drive assigned to the IFS of your system i to transfer the PTF from your PC to your system i. It is assumed that you have stored the PTF zip file (which your downloaded from the Internet or received via e-mail) on your PC.

Step 1

On your PC, create an install directory for the PTF, either within Windows Explorer, or from the DOS prompt, or use the directory you created for the original installation, e.g.:

```
md c:\PC_dir
```

where PC_dir is the name of the directory you wish to use for this installation.

Save or move the zip file that you downloaded or received into this directory.

Step 2

On your PC, unzip the contents of the zip file into the directory you just created using pkunzip, WinZip, Norton UnZip etc.

Step 3

On your PC, ensure that you have a network drive assigned to a directory in the root file system of your system i's IFS. This can be done in Windows Explorer or (My) Computer by selecting Tools, Map Network Drive.

Create a directory in the IFS to receive the PTF. This can be done in Windows Explorer by selecting File, New, Folder, or at the DOS prompt as follows:

```
md i:\IFS_dir
```

where **I:** is the network drive assigned to your system i and IFS_dir is the name of the directory you wish to use for this installation.

Step 4

Copy the contents of your PC's PTF directory created in Step 1 above to the IFS directory created in Step 3 above using Windows Explorer, or from the DOS prompt as follows:

```
copy c:\PC_dir\*.* I:\IFS_dir
```

where **PC_dir** is the name of the directory on your PC, **I:** is the network drive assigned to your system i and **IFS_dir** is the name of the directory you wish to use for this installation.

Step 5

On your system i, first sign off then sign back on again to ensure that you do to have any Slipstream objects in your QTEMP library. Then decide which library you wish to use for the install procedure and create a save file in that library. The save file should be called **Q5SLnnnn**, where **nnnn** is the PTF id, e.g.

```
crtsavf lib_name/Q5SLnnnn
```

where **lib_name** is the name of the library you chose to use

and **nnnn** is the PTF id.

Step 6

On your system i, copy the contents of the **Q5SLnnnn.sav** file (where **nnnn** is the PTF id) from the IFS directory to the save file just created, by issuing the following command:

```
cpyfrmstmf      fromstmf('/IFS_dir/Q5SLnnnn.sav')
                tombr('/qsys.lib/lib_name.lib/Q5SLnnnn.file')
                mbropt(*replace)
                cvtdta(*none)
                endlinfmt(*fixed)
                tabexpn(*no)
```

where **IFS_dir** is the name of the IFS directory into which you copied the PTF files in Step 4 above, and **lib_name** is the name of the library in which you created the save file in Step 5 above.

Step 7

On your system i, ensure that the previous step worked by displaying the contents of save file **Q5SLnnnn**:

```
dspsavf lib_name/Q5SLnnnn
```

If you see the message **(No records in save file)** then something has gone wrong during the preceding steps. Try following them again. If you still cannot get this procedure to work, contact us at **support@ariadnesoftware.co.uk.**

Step 8

On your system i, issue the following command to load the PTF on to your system:

```
lodptf          licpgm(5SLPSTM)
                dev(*savf)
                select(5SLnnnn)
                savf(lib_name/Q5SLnnnn)
```

where lib_name is the name of the library in which you created the save file in Step 3 and nnnn is the PTF id.

Step 9

On your system i, issue the following command to apply the PTF to your system:

```
apyptf          licpgm(5SLPSTM)
                select(5SLnnnn)
                apy(*temp)
```

where: **nnnn** is the PTF

If you have any trouble at any point in this procedure, please contact us by e-mail at **support@ariadnesoftware.co.uk.**

Where Did My Output Go?

Each of Slipstream's CVTDBFxxxx commands converts an iSeries spooled file to a stream file in a format such as Excel, XML or HTML. Where the output is created depends on what you specify on the TOSTMF parameter of the CVTDBFxxxx command that you ran. You have a number of options which we will discuss shortly.

Normally you will want to access these stream files from a PC application such as a spreadsheet application or a browser. How you access Slipstream output from your PC depends on a number of factors which we will also consider now.

The TOSTMF parameter

When you run one of the CVTDBFxxxx commands, you specify where you want the output to go and what you want it to be called on the **TOSTMF (To Stream File)** parameter.

There are 3 basic options:

- **IFS path name**

You can define an absolute or relative IFS path specifying the name of the file to be created and the directory in which it will be placed.

The IFS is a collection of file systems provided by your iSeries. Depending on which file system you select, your output may be stored locally on your iSeries' disks or remotely on another system on your network, which could be a PC, another iSeries a UNIX server etc.

Use of the IFS is explained more fully below.

The special value ***FROMFILE** (the parameter default value) tells Slipstream to create a file name from the name of the spooled file and an appropriate extension based on the format of the file being created (e.g. .xml for a XML file, .xls for an Excel file etc.) and place it in the current directory of the job.

- ***FTP**

This tells Slipstream to send the output using FTP (File Transfer Protocol) to another system running an FTP server process. This could be another iSeries, a PC server, a UNIX machine etc.

- ***EXITPGM**

This indicates that you will specify the location at a later stage in an exit program that will be called while Slipstream is running.

Understanding IFS path names

The IFS (Integrated File System) is a collection of file systems that your iSeries can use to store and retrieve information. Depending on which file system you choose to use, the data may be stored locally (on your iSeries' own disks) or remotely (on another system in your network).

When you enter a path name on the TOSTMF parameter, you are telling Slipstream the name of the file you wish to create. You will also be telling it, explicitly or implicitly, in which file system and directory to save that file.

The path consists of four elements:

- **The Extension**

If you type a name that ends with a period (.) and then a sequence of characters, you have specified an **extension**.

For example: **.xml, .xls, .htm**

Windows and other operating systems may use this extension to determine what type of file you have created. For example, if you double-click in Windows on a file name ending in .xls, it is likely that Windows will start or switch to Microsoft Excel and open the file.

This makes it very important that you should choose an extension which is appropriate to the type of file you are creating.

For example, if you are using CVTDBFXLS to create am=n Excel file, specify a file name ending in **.xls** so Windows recognizes that the file should be opened with Excel, but if you are using CVTDBFHTML to create an HTML file, choose a file name ending in **.htm** to ensure that Windows recognizes the file as HTML and will open it in your browser.

- **The File Name**

The part of the path name that precedes the extension is the name of the file itself. Slipstream does not impose any restrictions other than the limit of 1,024 bytes for the entire path name.

Please note, however, that the syntax and rules that apply to the name will be dependent on the file system you choose. For example, the QDLS file system (“shared folders”) does not allow the file name to be longer than 8 characters with an optional extension of 1-3 characters (old DOS-style 8.3 naming). Also note file names in some file systems are case-insensitive (e.g. root file system) while file names in other file systems are case-sensitive (e.g. QOpenSys).

- **The Directory Path**

You can optionally specify a directory or list of sub-directories in which the file is to be saved.

For example, if you have a directory called sales with subdirectories for each region, and then subdirectories for each year and month, you may need to specify a path such as:

sales/north/2009/nov

to indicate that the directory in which you wish to save you file is the November subdirectory within the 2009 subdirectory of the north region’s subdirectory within **sales**.

- **The File System**

You can optionally specify a file system name at the beginning of the path to indicate to which file system the path refers.

Here is a list of commonly used file system names that can be used at the beginning of a path name. Note that each begins with a / (forward slash) and that the root file system is indicated by a single forward slash alone:

/	The “root” file system. This is the “default” iSeries hierarchical file system.
/QDLS	Document Library Services (“shared folders”)
/QNTC	Windows NT Server file system. This file system provides access to data and objects that are stored on a Windows server **This file system can be used to directly read data from and write data to a separate Windows server on your network.**

/QOpenSys	A hierarchical file system compatible with UNIX and POSIX. Uses case-sensitive names.
/QSYS.LIB	The iSeries database. Although it is possible to save Slipstream output in a database file member, this is not recommended as the data is unlikely to be easily accessed there.

You should also understand the difference between an **absolute** path name and a **relative** path name.

An absolute path name is one which explicitly defines the full location at which a file is to be saved.

For example, the path name

/sales/north/2009/nov/new_business.xml

is an absolute path name which specifies the full location of a file to be created and breaks down as follows:

/	The initial / indicates the root file system
sales	The name of the directory in the root file system
north	The name of a subdirectory within **/sales**
2009	The name of a subdirectory within **/sales/north**
nov	The name of a subdirectory within **/sales/north/2009**
new_business	The name of the file to be created
.xml	The file extension, indicating an XML file

However, if you do not enter a forward slash (/) at the beginning of a path name, your iSeries will interpret this as a **relative** path name. Relative path names are interpreted relative to the current directory of the job (similar to the current directory in Windows or DOS).

For example, if your current directory is already set to **/sales**, the path

north/2009/nov/new_business.xml

(note there is no leading /) would be interpreted relative to **/sales** and would refer to exactly the same location as the absolute path

/sales/north/2009/nov/new_business.xml

The current directory of your job can be set with the **CHGCURDIR** or **CD** commands. Often, the current directory will be set automatically for you when you sign on to the iSeries based upon the HOMEDIR (home directory) attribute of your user profile.

Assume that your user profile has HOMEDIR = **/home/john**, indicating that when you sign on the current directory should be set to the **john** subdirectory within the **home** directory of the root file system. Unless you have changed this with CHGCURDIR or CD, if you specify a relative path name, the path will be interpreted relative to your current directory **/home/john.**

For example, the relative path

reports/sales.xml

would be interpreted as referring to a file called **sales.xml** in a subdirectory called reports within **/home/john**.

You will need to enclose path names in single quotes (‘) on the TOSTMF parameter if they contain forward slashes or other special characters.

For example:

TOSTMF(new_business.xml)

is acceptable to OS/400 without single quotes, but your iSeries will insist that:

TOSTMF(‘/sales/north/2009/nov/new_business.xml’)

is entered with single quotes around the path name. When prompting the command with F4, the iSeries will enclose the path name in quotes for you if you have not already done it.

Further information on the IFS can be found at:

http://publib.boulder.ibm.com/iseries/v5r2/ic2924/index.htm?info/ifs/rzaaxmst02.htm

Choosing where to store your output

When it comes to deciding where to save your Slipstream output, a number of factors need to be considered, for example:

- **Simplicity**

How easy is it to save files to and retrieve files from a particular IFS file system? Are the naming rules for the file system complex or restrictive?

- **Performance**

How well does that file system perform? Is saving and retrieving data from that file system quick and efficient or slow and laborious?

- **Reliability**

Will the file system always be available or is there a chance that it might be unavailable for some reason at the time when you try to save data to it or retrieve data from it?

- **Access**

What choices do you have with regards to accessing the data? How easy is it to retrieve data from the file system you choose to use using an appropriate application? For example, how easy is it to open a Excel file from a PC?

- **Management**

How easy is it to perform management functions on the files in the file system, such as backup, archiving and purging of old documents?

- **Security**

Can you ensure that only the right people have access to the documents?

- **Scalability**

Will problems occur when volumes increase?

We will now consider the various IFS file systems you are most likely to want to use according to these criteria.

Root File System

The “root” file system is in many ways the “default” IFS file system and is probably where most Slipstream users choose to store their output.

You save a Slipstream file in the root file system if you enter a path name on the TOSTMF parameter which does not explicitly and implicitly refer to any other file system.

Users can access files created on your iSeries in the “root” file system using network drives. For example, if your users have their I: drive assigned to the iSeries root file system, they could open a file called sales_report.xls saved in a directory called sales by opening i:/sales/sales_report.xls in Excel.

Simplicity	Excellent. The simplest and easiest to use. Long file names are supported. Not case-sensitive.
Performance	Good. Writing data locally will keep down the time taken to create the files. Speed of retrieval from a PC will depend on your network and other factors such as the power and loading of your iSeries.
Reliability	Excellent. Writing data locally means that file creation is not dependent on the availability of the network or another system.
Access	Good. Easy to access from Windows using network drives.
Management	Good. Can be backed up with the iSeries. Can be managed from the iSeries command line or from Windows using a network drive.
Security	Excellent. iSeries security applies.
Scalability	Moderate. High cost of iSeries disks a possible issue.
Comments	Recommended unless other factors dictate otherwise

QDLS File System

The QDLS or “shared folders” file system implements a DOS-style method of saving PC files and other documents on the iSeries own disks. It is really a legacy file system providing backwards compatibility for older applications written for the S/38 or versions of OS/400 that pre-date the availability of the IFS (OS/400 V3R1M0).

You save a Slipstream file in the QDLS file system if you enter a path name on the TOSTMF parameter which starts /QDLS or if you use a relative path name and your current directory path starts /QDLS.

Users can access files created on your iSeries in the QNTC file system using network drives. For example, if you users have their I: drive assigned to the iSeries root file system, they could open a file called REPORT.xml saved in a shared folder called SALES by opening i:/QDLS/SALES/REPORT.xml in Adobe Acrobat.

Simplicity	Good. Familiar to long-standing users of S/38 and AS/400 applications. Not case-sensitive. Naming limited to DOS-style 8.3 conventions so long file names will cause errors.
Performance	Poor compared to the “root” file system.
Reliability	Excellent. Writing data locally means that file creation is not dependent on the availability of the network or another system.
Access	Good. Easy to access from Windows using Network drives

Management	Good. Can be backed up with the iSeries. Can be managed from the iSeries command line or from Windows using a ISeries Access network drive.
Security	Excellent. iSeries security applies.
Scalability	Moderate. High cost of iSeries disks a possible issue.
Comments	Use the “root” file system instead.

QNTC File System

The QNTC file system is the iSeries implementation of Windows network neighborhood. It allows you to write to and read from files stored on a Windows server.

You save a Slipstream file in the QNTC file system if you enter a path name on the TOSTMF parameter which starts /QNTC or if you use a relative path name and your current directory path starts /QNTC. The file system name /QNTC should be followed by the name of the server, then the name of the shared resource on that server (e.g. the shared directory name) and then the path within that shared directory.

Imagine you have a Windows server which is known to the network as **server1**. On that server there is a directory called **sales** which is shared under the name **sales**. Within that shared directory there is a subdirectory called **2009**. If you have QNTC configured and your security settings allow it, you can save a file called november.xml in that subdirectory from the iSeries by specifying the path name:

/QNTC/server1/sales/2009/november.xml

The QNTC file system can be quite difficult to configure and manage, but once you have it running it can provide a very effective means of creating Slipstream output directly on a Windows server in your network.

Please note in particular that the iSeries user profile of the job which accesses QNTC must be the same name and have the same password as a user id that Windows networking recognizes.

Further information on QNTC is at:

http://publib.boulder.ibm.com/iseries/v5r2/ic2924/index.htm?info/ifs/rzaaxmstqntcfs.htm

http://www-1.ibm.com/support/docview.wss?uid=nas1aea450153eebf8ff8625670f0072550f&rs=110

http://www.itjungle.com/fhg/fhg031704-story04.html

http://www.itjungle.com/mgo/mgo111903-story02.html

Once you have saved your files on a Windows server in your network, users can then access files created with Slipstream on that Windows server using Windows networking. For example, if they have their F: drive assigned to a directory called sales on that server, they could access a file called sales_report.xml in that directory simply by opening file F:/sales_report.xml.

Simplicity	Can be difficult to set up and manage. Once files are saved on the Windows server, access should be very simple.
Performance	Creating files across the network on the PC server may be slow. Retrieval of files once created should be very fast but will depend on the server and network loading.
Reliability	Creating files across the network on the PC server requires both the

	server and the network to be available at the time.
Access	Easy to access from Windows using Windows networking.
Management	Good. Will need to be backed up with your Windows server.
Security	Good. Windows security applies.
Scalability	Good. Low-cost PC disks can be used.
Comments	If you prefer to store your files on a Windows PC server rather than on the iSeries, this is an ideal solution if the initial setup issues can be overcome and you can ensure that the PC server will be available to the iSeries when it needs to create the files.

Typical Solutions

When implementing Slipstream, it is important to make the right choices about where you will save the files you create and how you will access them.

Here are a few typical approaches that users have successfully implemented in the past.

- **Save the files in the iSeries "root"**

This is a really simple, easy and reliable method.

To save a file in the "root" file system, you just specify a path name starting with a forward slash /.

You can open files saved in the root file system from your PC applications (Acrobat, Excel, Word etc.) by using ISeries Access network drives to open the file just as you would a file saved locally on you PC or on a Windows or UNIX server.

The only real downside of this approach is that the files occupy space on your iSeries disks, which can be expensive compared to PC disks.

For further information on configuring the iSeries so your users can connect from their PC using iSeries Access network drives, see
publib.boulder.ibm.com/infocenter/iseries/v5r3/ic2924/info/rzaij/rzaijconnetas.htm

For further information on configuring your users' PC so they can connect to the iSeries using iSeries Access network drives, see
http://publib.boulder.ibm.com/infocenter/iseries/v5r3/ic2924/info/rzaij/rzaijnetserverpc.htm

- **Save the files directly to a Windows server using QNTC**

As explained above, the QNTC file system allows you to write directly to a Windows server from your iSeries.

Once QNTC is configured, you can use Slipstream to create your files on a suitable Windows server by specifying a path name starting /QNTC on the TOSTMF parameter of the Slipstream command you are running.

Once your files are saved on your Windows server, they can be accessed by any authorized user who can connect to that server.

- **Save the files directly to a Windows server using FTP**

As an alternative to using the QNTC file system, if your Windows server is running the FTP service, you can use the Slipstream TOSTMF(*FTP) option to send the output to that server via FTP.

Once your files are saved on your Windows server, they can be accessed by any authorized user who can connect to that server.

- **Email**

In the past you may have produced a large number of iSeries spooled files which were printed then distributed them on paper through your internal or external mail.

This process can be transformed into an automated, low-cost electronic service by creating XMLs, RTFs or Excel files from your spooled files rather than printing them on paper.

If you have installed CoolSpools Plus or ariadne's Communiqué email package (see www.ariadnesoftware.co.uk/Communique.htm), or if you have some other method of sending email from your iSeries, you can them distribute them electronically by email. The stream files could then be deleted once they had been emailed if they were no longer required.

Slipstream Variables

Certain parameters listed below support the use of Slipstream variables.

Slipstream variables consist of a pre-defined variable name from the list below enclosed in<: … :> (start of variable marker = left-hand angle bracket followed by a colon, end of variable marker = colon followed by a right-hand angle bracket).

You can define a different pair of markers from <: and :>by adding/changing the environment variables **SL_VAR_LEFT_MARKER** and **SL_VAR_RIGHT_MARKER.** For example, if you have SL_VAR_LEFT_MARKER set to $% and SL_VAR_IGHT_MARKER set to %$, you would use $%PAGSETNBR%$ rather than <:PAGSETNBR:> etc.

These variable names, including the markers, are replaced at run time by the corresponding data value. Variable names are not case-sensitive.

Variable name	Description
<:CURJOB:>	Current job name
<:CURUSER:>	Current user id
<:CURJOBNBR:>	Current job number
<:FROMFILE:>	Name of the database file specified on FROMFILE
<:FROMLIB:>	Library specified on FROMFILE
<:FROMMBR:>	Member name specified on FROMFILE
<:RCDFMT:>	Name of the record format in the file specified on FROMFILE. Invalid if more than one record format is selected for processing.
<:FILETEXT:>	File-level text of the file specified on FROMFILE
<:MBRTEXT:>	Member-level text of the member specified on FROMFILE
<:RCDTEXT:>	Record format-level text of the record format in the file specified on FROMFILE. Invalid if more than one record format is selected for processing.
<:QRYDFN:>	Name of the Query/400 query object specified on the QRYDFN parameter
<:QRYDFNLIB:>	Library name specified on the QRYDFN parameter
<:QMQRY:>	Name of the QM Query object specified on the QMQRY parameter
<:QMQRYLIB:>	Library name specified on the QMQRY parameter
<:STMFEXT:>	Default file extension corresponding to the format being output (e.g. '.CSV' when CSV being generated or '.XLS' when an Excel file is being created.
<:TOFMT:>	To-format. The format of the data being generated (corresponding to the TOFMT parameter of the CVTDBFSTMF command), .e.g. '*CSV', '*XLS'

<:CURDATE:>	The current datei n the format of the current job (DATFMT attribute).
<:CURDATE*xxx:>	The current date in the format indicated by *xxx, where *xxx is any one of: *YMD, *MDY, *DMY, *YYMD, *MDYY, *DMYY, *CYMD, *CMDY, *CDMY, *ISO, *EUR, *JIS, *JUL, *LONGJUL, *JOB or *SYSVAL.
<:CURDAY:>	The current day of the month as a number 01-31.
<:CURMONTH:>	The current month as a number 01-12.
<:CURYEAR:>	The current year as a number 0001-9999
<:CURYEAR4:>	The current year as a number 0001-9999
<:CURYEAR3:>	The current year as a number c01-c99 where c is 0 for the 20th century and 1 for the 21st.
<:CURYEAR2:>	The current year as a number 01-99.
<:CURTIME:>	The current time in hhmmss format.
<:OWNUSER:>	The user profile that owns the spooled file.
<:CURUSEREMAIL:>	The email address of current user. The email address is the SMTP email address of the user from the system directory.
<:CURUSERNAME:>	The name of the current user. The name is derived from the information held for the user in the system directory.
<:CURUSERHOME:>	The home directory of the current user. The home directory is taken from the HOMEDIR attribute of the user profile.

Example:

```
CVTDBFXLS
    FROMFILE(QADSPOBJ)
    TOSTMF('<:fromlib:>_<:fromfile:>_<:frommbr:>.xls')
    EMAIL(*YES)
    EMAILTO((<:CURUSEREMAIL:>))
```

Here the CVTDBFXLS command is being applied to a database file called QADSPOBJ. The name of the stream file to be generated will be derived from various the library name, file name and member name, e.g.:

QGPL_QADSPOBJ_QADSPOBJ.XLS

Slipstream Functions

When using Slipstream variables, you can also use a number of Slipstream functions to adjust the data substituted at run time for each variable. These functions can often be helpful in converting the data returned by a variable to a consistent, standard format. For example, you might want to use Slipstream variables to build the names of the PDF files you're creating from data held inside the spooled file. Slipstream functions can help with this, for example by allowing you to:

- remove any leading or trailing spaces
- pad numeric value to a constant fixed length with leading zeros

- translate certain characters which would be invalid in a file name (such as /) to an alternative acceptable character (such as -)

By default, Slipstream functions consist of a pre-defined function name from the list below preceded by the marker **$$** but You can define a different marker from $$ by adding/changing the environment variable **SL_FCN_MARKER.** For example, if you have SL_FCN_MARKER set to %%, you would use %%TRIM, %%PADL etc. rather than $$TRIM, $$PADL etc.

Function parameters are enclosed in parentheses () and separated by commas. Character values used as parameters are case-sensitive and can be either enclosed in single quotes ' ' (doubled up where required by OS/400), double quotes " ", or not enclosed by anything.

Function names are not case-sensitive.

Function name	**$$TRIM**
Description	Trim characters from the left and right sides of the data. Similar to the ILE RPG %trim() builtin function.
Parameters	
1	Data to trim (typically a Slipstream variable).
2	Characters to remove (optional, default = blank).
Examples	
$$TRIM(<:FILETEXT:>)	Trims blanks from the start of the value returned by Slipstream variable <:FILETEXT:>. For example, the value “ 000123.45- “ becomes “ 000123.45-“
$$TRIM(<:FILETEXT:>,’0’)	Trims zeros from the start of the value returned by Slipstream variable <:FILETEXT:>. For example, the value “000123.4500 “ becomes “123.45 “.

Function name	**$$TRIML**
Description	Trim characters from the left (start) of the data. Similar to the ILE RPG %triml() builtin function.
Parameters	
1	Data to trim (typically a Slipstream variable).
2	Characters to remove (optional, default = blank).
Examples	

$$TRIML(<:FILETEXT:>)	Trims blanks from the start of the value returned by Slipstream variable <:FILETEXT:>. For example, the value “ 000123.45- “ becomes “000123.45- “
$$TRIML(<:FILETEXT:>,’0’)	Trims zeros from the start of the value returned by Slipstream variable <:FILETEXT:>. For example, the value “000123.4500 “ becomes “000123.45 “.

Function name	**$$TRIMR**
Description	Trim characters from the right (end) of the data. Similar to the ILE RPG %trimr() builtin function.
Parameters	
1	Data to trim (typically a Slipstream variable).
2	Characters to remove (optional, default = blank).
Examples	
$$TRIMR(<:FILETEXT:>)	Trims blanks from the end of the value returned by Slipstream variable <:FILETEXT:>. For example, the value “ 000123.45- “ becomes “ 000123.45-“
$$TRIMR(<:FILETEXT:>,’0’)	Trims zeros from the end of the value returned by Slipstream variable <:FILETEXT:>. For example, the value “000123.4500 “ becomes “000123.45 “.

Function name	**$$PADL**
Description	Pad a string to a given length by adding a specified character at the start.
Parameters	
1	Data to pad (typically a Slipstream variable).
2	Length to pad to
3	Characters to pad with (optional, default = blank).

Examples	
$$PADL(<:FILETEXT:>,10)	Pads the value returned by Slipstream variable <:FILETEXT:> to a length of 10 characters by adding blanks at the start. For example, the value “123.45- “ becomes “ 123.45-“
$$PADL(<:FILETEXT:>,10,’0’)	Pads the value returned by Slipstream variable <:FILETEXT:> to a length of 10 characters by adding zeros at the start. For example, the value “123.45- “ becomes “0000123.45-“

Function name	**$$PADR**
Description	Pad a string to a given length by adding a specified character at the end.
Parameters	
1	Data to pad (typically a Slipstream variable).
2	Length to pad to
3	Characters to pad with (optional, default = blank).
Examples	
$$PADL(<:FILETEXT:>,10)	Pads the value returned by Slipstream variable <:FILETEXT:> to a length of 10 characters by adding blanks at the end. For example, the value “123.45-“ becomes “123.45- “
$$PADL(<:FILETEXT:>,10,’0’)	Pads the value returned by Slipstream variable <:FILETEXT:> to a length of 10 characters by adding zeros at the end. For example, the value “123.45“ becomes “123.450000 “

Function name	**$$SUBST**
Description	Returns a substring. Similar to ILE RPG’s %subst.
Parameters	

1	Data to substring (typically a Slipstream variable).
2	Start position
3	Length (optional, default = to end of string).
Examples	
$$SUBST(<:FILETEXT:>,5)	Returns the substring of the value returned by Slipstream variable <:FILETEXT:> starting at character position 5 and extending to the end of the string. For example, the value “0000123456“ becomes “123456 “.
$$PADL(<:FILETEXT:>,5,3)	Returns the substring of the value returned by Slipstream variable <:FILETEXT:> starting at character position 5 and extending for 3 characters. For example, the value “0000123456“ becomes “123 “.

Function name	**$$UPPER**
Description	Converts a string to upper case, assuming the CCSID of the job.
Parameters	
1	Data to convert (typically a Slipstream variable).
Examples	
$$UPPER(<:FILETEXT:>)	Converts the value returned by Slipstream variable <:FILETEXT:> to upper case. For example, the value “John Smith“ becomes “JOHN SMITH“.

Function name	**$$XLATE**
Description	Translates characters in the data. Similar to the ILE RPG %xlate function.
Parameters	
1	List of characters to translate from
2	List of characters to translate to

3	Data to translate (typically a Slipstream variable).
4	Start position (optional, default = first)

Examples	
$$XLATE(“ “,”_”,<:FILETEXT:>,1)	Translates spaces in the value returned by Slipstream variable <:FILETEXT:> to underscores, starting at the first character. For example, the value “John Smith“ becomes “John_Smith“.

Excel Placeholders

When specifying header and footer text to be printed with an Excel file (XLSPRINT parameter), you can use Excel placeholders. Excel will substitute a data value for the placeholder before printing the file.

Note that while Slipstream variables are defined by Slipstream and a data value is substituted for them by Slipstream as it creates the file, Excel placeholders are substituted by Excel as it prints the file. So, if you were to include the current date in an Excel footer text using a Slipstream variable such as <:CURDATE:>, that value would be replaced by the current date at the time the file was created and would not change when the file is printed, whereas if you use an Excel placeholder such as &D to insert the current date, that date will be substituted by Excel with the current date every time the file is printed.

Excel placeholders take the form of a single letter preceded by an ampersand (&), with the sole exception of the codes indicating a change of font name and/or size (see below).

&P	Current page number
&N	Page count
&D	Current date
&T	Current time
&A	Sheet name
&F	File name without path
&Z	File path without file name (BIFF8)
&U	Underlining on/off
&E	Double underlining on/off
&S	Strikeout on/off
&X	Superscript on/off
&Y	Subscript on/off
&"<fontname>"	Set new font <fontname>
&"<fontname>,<fontstyle>"	Set new font with specified style <font style>. The style <fontstyle> is in most cases one of "Regular", "Bold", "Italic", or "Bold Italic". But this setting is

dependent on the font, it may differ (localised style names, or "Standard", "Oblique", ...)

&<fontheight> Set font height in points (<fontheight> is a decimal value).

If this command is followed by a plain number to be printed in the header, it must be separated from the font height with a space.

Example: &"Arial,Bold Italic"&14Page &P of &N

On Page 2 of a 3-page spreadsheet, this header/footer string prints the text "Page 2 of 3" in Arial Bold Italic 14-point font.

Using Slipstream

Getting Started

Where **product_library** is shown below, substitute the actual product name from this table:

If running the standalone version of Slipstream	**SLPSTMV5R2**
If running CoolSpools PLUS	**COOLSPV5R2**

You run the main Slipstream functions by executing one of the format-specific commands on your system i.

Either:

(a) prompt one of the following command strings with F4:

To create an excel file:	**product_library/CVTDBFXLS**
To create an XML file:	**product_library/CVTDBFXML**
To create a delimited ASCII file (e.g. CSV):	**product_library/CVTDBFCSV**
To create an HTML file:	**product_library/CVTDBFHTML**
To create a fixed-width text file:	**product_library/CVTDBFTXT**
To extract cells contents from Excel files:	**product_library/CVTXLSDBF**
To display a password in encrypted form	**product_library/DSPENCPWD**
To zip (compress) a file or files	**product_library/ZIPDTA**
To unzip (decompress) a file or files	**product_library/UNZIPDTA**

Or

(b) display the Slipstream menu by entering: **GO product_library/SLIPSTREAM**

Using styles

The new DFNSTYLES parameter lets you defines styles to be applied to your output when you are converting to Excel, HTML or XML formats. These styles specify the appearance of data on screen when the spreadsheet is opened (in MS Excel or another spreadsheet application) or when the HTML or XML document is opened (in your browser).

There are five predefined styles which define the default appearance of different types of information in Excel and HTML:

Style Name	**Description**
*DATA	Data rows
*HEADER	Column heading rows generated based on the setting of the HEADER parameter
*TITLE	Additional heading rows (additional headings defined on HEADER plus HTML caption)

*SUBTOTAL	Subtotals in Query/400 output using the *COMBINED output type
*TOTAL	Final totals in Query/400 output using the *COMBINED output type

plus two more that are relevant only to XML output:

Style Name	**Description**
*ROOT	The root element of the document
*ROW	The row element of the document. The row element is that corresponding to records in the input file.

If you do not specify these styles on the DFNSTYLES parameter, they will assumes certain default values (see the table in the section on the DFNSTYLES parameter below). On the other hand, if you do define one or more of these styles on the DFNSTYLES parameter, the values you enter for that parameter override the defaults for the type of information associated with that style.

In addition, you can define your owned named styles. You can then use the FLDSTYLES parameter to associate your named style with one or more individual fields in the input file.

Example:

CVTDBFXLS

...

DFNSTYLES((HIGHLIGHT *YES *NO *GENERAL *NONE *BOTTOM *NO *NO *AUTOFIT *ARIAL 12 *YES *NO *NO *YELLOW *BLUE *AUTO *NONE *THIN))
FLDSTYLES((BALANCE *FLDNAM HIGHLIGHT))

This code defines a new style called “highlight” that uses Arial bold 12-point yellow on blue and applies that style to the field called “BALANCE”.

Using conditional formatting

Styles are also used when you want to apply conditional formatting rules to Excel spreadsheets that Slipstream generates. Conditional formatting lets you modify the appearance of cells in the spreadsheet depending on whether certain rules you define are met or not. For example, if your spreadsheet contains data from customer accounts, you might color those rows that relate to accounts with a negative balance red to highlight them, while those with a credit balance over $1,000 might be colored green.

Use the CNDFMTGRP (Conditional Formatting Groups) parameter to define the range of columns to which a group of related rules should be applied.

Use the CNDFMTRULE (Conditional Formatting Rules) parameter to define the rules to be applied and the style (as defined on DFNSTYLES) that will be used to format cells where those rules evaluate to true.

Using encrypted passwords

In the past, if you specified a password on a command such as CVTDBFSTMF and embedded that command in your CL source code, you would need to store that password in plain text form. This was clearly a security exposure.

Now, Slipstream gives you the opportunity to use encrypted passwords on all command parameters that accept a password string. An encrypted password is a scrambled version of your password which is returned to you when you supply the actual password to the DSPENCPWD (Display Encrypted Password) command. You can then code the scrambled password in your source code and specify *YES for the associated **“Encrypted password supplied”** element to indicate to Slipstream that it needs to decrypt the password before use.

For example, if you supply the password “test” to DSPENCPWD, thus:

DSPENCPWD PWD('test')

it send you the completion message:

Encrypted password is X'178D2D35E0EBFF508A63252433D6C4E0'.

You can then use this encrypted password on commands that require a password, e.g.:

ZIPDTA ... PWD(X'178D2D35E0EBFF508A63252433D6C4E0' *YES)

The password of the zipped file(s) will be “test”.

CVTDBFxxxx Command Parameters

The sections below look at each of the parameters to the various ***CVTDBFxxxx*** commands (CVTDBFXLS, CVTDBFXML, CVTDBFHTML, CVTDBFPDF, CVTDBFCSV, CVTDBFTXT, plus the now deprecated CVTDBFSTMF) in turn and explain how they should be used to get the most out of this invaluable utility.

In the examples, an ellipsis (…) indicates that a number of required parameters have been omitted for the sake of clarity.

FROMFILE – From database file

Parameter	**FROMFILE**
Description	**Specifies the database file to be converted or indicates the source of the data to be converted (query, SQL)**
Applies to commands:	**CVTDBFXLS, CVTDBFCSV, CVTDBFHTML, CVTDBFPDF, CVTDBFTXT, CVTDBFXML, CVTDBFPDF** **CVTDBFSTMF (deprecated)**
Dependent on:	**None**
Supports Slipstream variables	**No**

The **FROMFILE** (From File) parameter specifies the name and library of the database file that contains the records to be converted. Alternatively, one of the special values explained below can be used to run a query and save the output to a stream file as a single operation.

A database file can be a physical file or a logical file.

There are three elements to this parameter.

From File

The first element of the FROMFILE parameter is the qualified name of the file which contains the data to be converted or one of the special values listed below.

Unless one of the special values listed below is input, the file name must be specified as:

file-name	Specify the name of the database file that contains the records to be converted.

The possible library values are:

***LIBL**	All libraries in the user and system portions of the job's library list are searched until the first match is found.
***CURLIB**	The current library for the job is used to locate the database file. If no library is specified as the current library for the job, the QGPL library is used.
library-name	Specify the name of the library to be searched.

Special values for the FROMFILE parameter are as follows:

***SQL**	Run a SQL statement. The **SQL** parameter will be used to define the SQL statement to be run and the naming convention to be applied. The SQL statement will be validated, and, if valid, it will be executed and the output saved in a temporary result table, which will then be converted to the stream file format defined on the **TOFMT** parameter.
***SQLSRC**	Run a SQL statement held in a source file. The **SQLSRC** parameter will be used to specify the name of the source file and member where the SQL statement is held. The SQL statement will not be validated until it is executed. ***Slipstream*** will create a temporary Query Management (QM) query object from the contents of the source file. This temporary QM query will be executed and the output saved in a temporary result table, which will then be converted to the stream file format defined on the **TOFMT** parameter.
***QMQRY**	Run a Query Management query. The **QMQRY** parameter will be used to specify the name of the QM query object to be run. ***Slipstream*** will execute the QM query and save the output in a temporary result table, which will then be converted to the stream file format defined on the **TOFMT** parameter.
***QRYDFN**	Run a Query/400 query. The **QRYDFN** parameter will be used to specify the name of the Query/400 query object to be run, and optionally, the file to be queried. ***Slipstream*** will execute the Query/400 query and save the output in a temporary result table, which will then be converted to the stream file format defined on the **TOFMT** parameter.

Select records and fields

The second element of the FROMFILE parameter allows you to control the prompting of record format names and field names. This prompting cannot be performed if one of the special values listed above has been specified for the file name parameter.

***NO**	(Default). Record format and field names will not be prompted.
***YES**	You will be prompted with a list of record format and field names from the file specified on the FROMFILE parameter, allowing you to indicate which fields and record formats should be included in the stream file, and in what order.

To prompt the command, type the command name (CVTDBFXLS etc.) on a command line and press F4.

From member

The third element of the FROMFILE parameter allows you to specify the name of the member in the file from which input will be taken.

This element must be ***FIRST** if one of the special values is specified for the file name.

***FIRST**	(Default). The first member is used.
Member_name	Specify the member to be used.

Example: CVTDBFXLS FROMFILE(CUSTLIB/CUSTFILE *YES)…

Here the CVTDBFXLS command is being applied to a database file called **CUSTFILE** which is located in library **CUSTLIB**. If you prompt the command with F4, you will be shown a list of fields and record formats from the file on the INCLFLD (Include Fields) and RCDFMT (Record Formats) parameters respectively.

Example: CVTDBFXLS FROMFILE(*SQL) …

Here the CVTDBFSTMF command is being used to run a piece of SQL. The SQL will be syntax checked, and, if valid, executed. The results will be saved in a temporary result table then converted to a stream file in the format specified on the **TOFMT** parameter.

TOSTMF – To stream file

Parameter	**TOSTMF**
Description	**Specifies the path name of the stream file to create or update or the special value *FTP indicating FTP output**
Applies to commands:	**CVTDBFXLS, CVTDBFCSV, CVTDBFHTML, CVTDBFPDF, CVTDBFTXT, CVTDBFXML, CVTDBFPDF** **CVTDBFSTMF (deprecated)**
Dependent on:	**None**
Supports Slipstream variables	**Yes**

The **TOSTMF** (To Stream File) parameter specifies the path name of the stream file into which converted data is placed. All directories in the path name must exist.

New directories are not created. If the stream file does not exist, it is created.

The name you enter here may be up to 1,024 characters long. However, the name that you choose must be a valid name for the IFS file system into which the stream file is to be created. For example, the shared folders (QDLS) file system only supports file names in the 8.3 format, i.e. a file name up to 8 characters long followed by an optional extension of up to 3 characters. If you choose an invalid file name, an error will occur and the file will not be saved.

You should choose a file name which is suitable for the type of file being created. For example, Excel files should be given the extension **.xls** so that they are recognized as Excel files by applications such as Excel and Lotus 123. HTML files are normally given as extension of **.htm** or **.html**. ASCII fixed text files often have an extension such as **.txt**, **.asc**, **.prn** or **.dat**.

Options are:

***FROMFILE**	(Default). A stream file name is generated based on the name specified on the FROMFILE parameter. The stream file will be created in the current directory. The stream file name will be the name of the file specified on the FROMFILE parameter followed by a period (.) and an appropriate extension indicating the type of file being created.
***FTP**	The output will be sent by FTP to a remote server running an FTP service (e.g. another system i, a UNIX machine or a Windows server). The output will not be stored permanently on the local system i. You will be required to define the details of the FTP server to receive the output on the FTP parameter.
to-stmf	Specify the full relative or absolute path name of the stream file to be created or updated.

For more information on specifying path names, refer to the IFS Introduction manual in the system i Information Center at http://publib.boulder.ibm.com/system i/V5R2/ic2924/info/ifs/rzaaxmstpath.htm#HDRPATH

Example: CVTDBFXLS FROMFILE(CUSTFILE) TOSTMF(*FROMFILE)

Here the CVTDBFXLS command is being applied to a database file called **CUSTFILE**. The data is to be converted to Excel format and, since **TOSTMF(*FROMFILE)** is specified, the stream file will be created in the current directory and will be called **CUSTFILE.XLS**.

Example: CVTDBFXLS FROMFILE(CUSTFILE) TOSTMF('/customer_data/customers.xls')

Here again the CVTDBFXLS command is being applied to a database file called **CUSTFILE.** This time the stream file to be created is called **customers.xls** and will be created in directory **customer_data** in the root file system.

TOFMT – To format

Parameter	**TOFMT**
Description	**Specifies the format of the output**
Applies to commands:	**CVTDBFSTMF (deprecated)**
Dependent on:	**None**
Supports Slipstream variables	**No**

The **TOFMT** (To Format) parameter allows you to select the format to which the database file should be converted.

Starting from this release (V5R2M0), Slipstream supplies a series of format-specific commands such as CVTDBFXLS (Convert Database File to Excel) and CVTDBFHTML (Convert Database File to HTML) where the format of the output is implicit in the command itself. This parameter therefore relates only to the CVTDBFSTMF command which can output several different formats. Use of the CVTDBFSTMF command is now deprecated. CVTDBFSTMF will not be enhanced in future releases and may be withdrawn in a future release.

The options are:

***XLS** (Default). Excel © (BIFF) format. The contents of the database file are converted to columns and rows in an Excel spreadsheet. Column widths are selected appropriately and formatting is implemented to reflect edit codes and edit words associated with the fields in the database file.

***CSV** Delimited ASCII text (e.g. Comma Separated Variable). The database file contents are converted to delimited ASCII text records. The delimiters used to separate fields and enclose strings are defined on the **CSV** parameter. This option enables you to create files in CSV (comma-separated variable) format and other similar formats (e.g. tab separated or blank separated) for loading into spreadsheets and other PC applications.

***HTML** HTML. The contents and attributes of the database file are converted to HTML format, suitable for viewing in a browser such as Internet Explorer or Netscape Navigator.

***FIXED** Fixed-format ASCII text. The contents of the database file are converted to ASCII text records. Each field is assigned a fixed width so that data for each field always

starts and ends in the same position. Each column is separated from the next by a single blank.

Example: CVTDBFSTMF FROMFILE(ORDERS)…
TOFMT(*XLS)

Here the CVTDBFSTMF command is being applied to a database file called **ORDERS** in order to create a stream file in Excel format.

STMFOPT – Stream file option

Parameter	**STMFOPT**
Description	**Indicates what action should be taken if the output file already exists.**
Applies to commands:	**CVTDBFXLS, CVTDBFCSV, CVTDBFHTML, CVTDBFPDF, CVTDBFTXT, CVTDBFXML, CVTDBFPDF** **CVTDBFSTMF (deprecated)**
Dependent on:	**None**
Supports Slipstream variables	**No**

The **STMFOPT** (Stream File Option) parameter allows you to select the action the command should take if the stream file you have specified on the **TOSTMF** parameter already exists. This parameter is ignored if the stream file does not already exist.

On the format-specific commands (e.g. CVTDBFXLS) there are three elements to this parameter. On CVTDBFSTMF there are only 2 elements.

Option

The first element specifies the option to betaken if the file exists.

The options are:

***NONE**	The command reports an error and the existing file is not changed. For safety's sake, this is the default value.
***REPLACE**	The existing file is replaced.
***ADD**	The contents of the database file are appended to the end of the existing file. In relation to Excel output, this results in a new worksheet being added.
***UNIQUE**	Slipstream generates a unique file name for the output file by appending a numeric suffix to the name specified on the TOSTMF parameter (before any extension). The numeric suffix will be one higher than the highest suffix associated with any existing file of this name in the directory. If a value other than *NONE is specified on the ***UNIQUE separator character** element below, that character is inserted between the name and the suffix.
***RPLXLSSHT**	When outputting to Excel format, this option allows you to specify that the new worksheet(s) created this time will replace one or more existing worksheets in the file. The worksheets to be replaced are specified on the RPLXLSSHT parameter.

Create Empty File

The second element specifies the option to be taken if there is no data to be output.

The options are:

***NO**	Slipstream produces no output if there is no data to convert.
***YES**	Slipstream produces an empty file if there is no data to convert (e.g. an Excel file with zero data rows).

***UNIQUE suffix separator**

On the CVTDBFSTMF command, this element does not exist on STMFOPT and the information is supplied instead on the separate **SEPCHAR** (Separator Character) parameter. It enables you to specify the separator character that Slipstream will use when generating a unique stream file name in conjunction with STMFOPT(*UNIQUE).

When STMFOPT(*UNIQUE) is specified, this character is inserted between the file name you specify on the **TOSTMF** parameter (minus the extension) and the numeric suffix which Slipstream appends to that name (minus the extension) to create a unique file name.

On the format-specific commands CVTDBFXLS etc., this parameter is now part of the STMFOPT parameter.

Options are:

***NONE**	(The default). No separator is used.
***UNDERSCORE**	An underscore character (_).
Any other character	Any other character allowed in a file name.

Example: ***CVTDBFXLS*** ***FROMFILE(INVOICES)***
TOSTMF(invoices.xls)
STMFOPT(*ADD)

Here the CVTDBFXLS command is being applied to a database file called **INVOICES** and converted to a stream file called **invoices.xls**. A new worksheet will be added to the existing file.

RPLXLSSHT – Replace Excel worksheet names

Parameter	**RPLXLSSHT**
Description	**Specifies the names of Excel worksheets to be replaced when STMFOPT(*RPLXLSSHT) is specified**
Applies to commands:	**CVTDBFXLS** **CVTDBFSTMF (deprecated)**
Dependent on:	**STMFOPT(*RPLXLSSHT)**
Supports Slipstream variables	**No**

When outputting to Excel format, specifying STMFOPT(*RPLXLSSHT) allows you to indicate that the new worksheet(s) created this time will replace one or more existing worksheets in the file. The worksheet(s) to be replaced is/are specified on this RPLXLSSHT parameter.

There is a single value:

***NO** (Default) No worksheet(s) will be replaced.

Alternatively, specify between 1 and 10 names of existing worksheets in the file specified on the TOSTMF parameter that will be replaced by the new worksheet(s) created this time the command is run. The new worksheet(s) will be inserted into the file after the worksheet preceding the first worksheet listed on this parameter or at the beginning of the file is the first worksheet in the file is listed here. All worksheets listed here will be dropped from the file.

Options are:

***FIRST**	The first worksheet
***LAST**	The last worksheet
worksheet_name	Specify the name of the worksheet.

FORMAT – Format specification

Parameter	**FORMAT**
Description	**Indicates the format of the data to convert where the DDS of the file is not the best guide**
Applies to commands:	**CVTDBFXLS, CVTDBFCSV, CVTDBFHTML, CVTDBFPDF, CVTDBFTXT, CVTDBFXML, CVTDBFPDF** **CVTDBFSTMF (deprecated)**
Dependent on:	**None**
Supports Slipstream variables	**No**

The **FORMAT** (Format specification) parameter lets you specify the name of a second file which will determine the structure of the data in the file being converted. This option is similar to the FORMAT parameter of the OPNQRYF command and can be useful where the data is held in a flat file with no DDS. Define the structure of the data in DDS, create an format file and specify that file on the FORMAT parameter.

Options are:

***FROMFILE**	(The default). The DDS (metadata) of the file specified on the FROMFILE parameter determines the structure of the data being converted.
Qualified_file_name	Specify the name of the file that will define the structure of the data being converted.

FROMRCD – From record number

Parameter	**FROMRCD**
Description	**The first relative record number to convert**
Applies to commands:	**CVTDBFXLS, CVTDBFCSV, CVTDBFHTML, CVTDBFPDF, CVTDBFTXT, CVTDBFXML, CVTDBFPDF** **CVTDBFSTMF (deprecated)**
Dependent on:	**None**
Supports Slipstream variables	**No**

The **FROMRCD** (From record) parameter identifies the relative record number of the first record to be converted.

Options are:

***START**	(The default). Conversion begins with the first record in the file.
Record_ number	The relative record number of the first record to be converted.

TORCD – To record number

Parameter	**TORCD**
Description	**The last relative record number to convert**
Applies to commands:	**CVTDBFXLS, CVTDBFCSV, CVTDBFHTML, CVTDBFPDF, CVTDBFTXT, CVTDBFXML, CVTDBFPDF** **CVTDBFSTMF (deprecated)**
Dependent on:	**None**
Supports Slipstream variables	**No**

The **TORCD** (To record) parameter identifies the relative record number of the last record to be converted.

Options are:

***END**	(The default). Conversion ends with the last record in the file.
Record_ number	The relative record number of the last record to be converted.

SEPCHAR – Separator character

Parameter	**SEPCHAR**
Description	**Separator character to use when building file names for STMFOPT(*UNIQUE)**
Applies to commands:	**CVTDBFSTMF (deprecated)**
Dependent on:	**STMFOPT(*UNIQUE)**
Supports Slipstream variables	**No**
Migration notes	**This parameter has been deleted and incorporated into the STMFOPT parameter on the new format-specific commands.**

The **SEPCHAR** (Separator Character) parameter enables you to specify that separator character that Slipstream will use when generating a unique stream file name in conjunction with STMFOPT(*UNIQUE).

When STMFOPT(*UNIQUE) is specified, this character is inserted between the file name you specify on the **TOSTMF** parameter (minus the extension) and the numeric suffix which Slipstream appends to that name (minus the extension) to create a unique file name.

On the format-specific commands CVTDBFXLS etc., this parameter is now part of the STMFOPT parameter.

Options are:

***NONE**	(The default). No separator is used.
***UNDERSCORE**	An underscore character (_).
Any other character	Any other character allowed in a file name.

SQL – SQL statement options

Parameter	**SQL**
Description	**The SQL statement to run and SQL statement options**
Applies to commands:	**CVTDBFXLS, CVTDBFCSV, CVTDBFHTML, CVTDBFPDF, CVTDBFTXT, CVTDBFXML, CVTDBFPDF** **CVTDBFSTMF (deprecated)**
Dependent on:	**FROMFILE(*SQL)**
Supports Slipstream variables	**Yes**

The **SQL** (SQL Statement Options) parameter only appears if **FROMFILE(*SQL)** is specified, indicating that the input is to be the result set of an SQL statement. This parameter allows you to input the SQL statement to be run and define the naming convention to be used.

There are 2 elements to the parameter.

SQL Statement

Input the SQL statement you wish to execute. This must be a SELECT statement, otherwise an error will occur. The statement will be syntax checked prior to being executed.

Naming

Specifies the naming convention used for objects in SQL statements.

***SYS**	(Default). The system naming convention (library-name/file-name) is used.
***SQL**	The SQL naming convention (collection-name.table-name) is used.

Example: CVTDBFXLS FROMFILE(*SQL) …
SQL(‘select * from custfile where state = ‘’NY’’’)

This example shows an SQL statement being run to provide the input to the CVTDBFXLS command. The output from the SQL query will be saved as an Excel file.

See the QRYSLT parameter for details of how that parameter relates to record selection performed with the SQL.

See the SORT parameter for details of how that parameter relates to record ordering performed with the SQL.

SQLSRC – SQL source options

Parameter	**SQLSRC**
Description	**The source member containing SQL to run and related options**
Applies to commands:	**CVTDBFXLS, CVTDBFCSV, CVTDBFHTML, CVTDBFPDF, CVTDBFTXT, CVTDBFXML, CVTDBFPDF** **CVTDBFSTMF (deprecated)**
Dependent on:	**FROMFILE(*SQLSRC)**
Supports Slipstream variables	**No**

The **SQLSRC** (SQL Source Options) parameter only appears if **FROMFILE(*SQLSRC)** is specified, indicating that the input is to be the result set created by running an SQL statement held in a source file. This parameter allows you to indicate the source file and source member where the SQL statement is stored.

Please note that:

- the file specified must exist
- the member specified must exist in the file
- the file must be a source physical file
- the source member must not be empty

There are 2 elements to the parameter.

SQL Source file

The first element of the **SQLSRC** parameter is the qualified name of the file which contains the SQL statement to be executed.

file-name Specify the name of the source physical file that contains the SQL statement to be executed.

The possible library values are:

***LIBL** All libraries in the user and system portions of the job's library list are searched until the first match is found.

***CURLIB** The current library for the job is used to locate the database file. If no library is specified as the current library for the job, the QGPL library is used.

library-name Specify the name of the library to be searched.

SQL Source member

The second element of the **SQLSRC** parameter is the name of the source member which contains the SQL statement to be executed.

member-name	Specify the name of the source member that contains the SQL statement to be executed.

Example: CVTDBFXLS FROMFILE(*SQLSRC) …
SQLSRC(SQLSRC CUSTSQL)

This example shows an SQL statement stored in a source file being run to provide the input to the CVTDBFXLS command. The output from the SQL query will be saved as an Excel file. The SQL statement is in source member CUSTSQL in source file SQLSRC.

See the QRYSLT parameter for details of how that parameter relates to record selection performed with the SQL.

See the SORT parameter for details of how that parameter relates to record ordering performed with the SQL.

QRYDFN – Query/400 options

Parameter	**QRYDFN**
Description	**Query/400 options when FROMFILE(*QRYDFN) used**
Applies to commands:	**CVTDBFXLS, CVTDBFCSV, CVTDBFHTML, CVTDBFPDF, CVTDBFTXT, CVTDBFXML, CVTDBFPDF** **CVTDBFSTMF (deprecated)**
Dependent on:	**FROMFILE(*QRYDFN)**
Supports Slipstream variables	**No**
Migration notes	**The Query File and Query File Member elements of this parameter have been deleted from the new format-specific commands. If you have code which references this parameter, it will need to be modified when moving away from CVTDBFSTMF to the new format-specific commands if that code references these deleted elements or any later elements.** **The Query File and Query File Member elements are now replaced by the QRYFILE parameter (which previously provided an alternative means of specifying a query file and was the only way of specifying multiple query files).**

The **QRYDFN** (Query/400 Options) parameter only appears if **FROMFILE(*QRYDFN)** is specified, indicating that the input is to be the output created by running a Query/400 query. This parameter allows you to indicate the name of the Query/400 query object to be run.

There are five elements to this parameter (seven in the case of CVTDBFSTMF):

Query/400 object

Specifies the qualified name of the Query/400 query object.

query-name Specify the name of the Query/400 query object (object type *QRYDFN) which should be run.

The possible library values are:

***LIBL** All libraries in the user and system portions of the job's library list are searched until the first match is found.

***CURLIB** The current library for the job is used to locate the database file. If no library is specified as the current library for the job, the QGPL library is used.

library-name Specify the name of the library to be searched.

Record selection

Allows you to control whether you will be prompted with the Query/400 Record Selection screen. This is only possible if the command is being run interactively and if Query/400 is licensed.

See the QRYSLT parameter for details of how that parameter relates to record selection performed with the SQL.

See the SORT parameter for details of how that parameter relates to record ordering performed with the SQL.

Options are:

***NO**	(Default). The Query/400 record selection screen is not displayed.
***YES**	The Query/400 record selection screen is displayed, if the command is run interactively.

Query file

This element has been deleted from the format-specific commands: use the QRYFILE parameter instead.

Specifies the qualified name of the file the Query/400 query should process as its input file.

***QRYDFN**	(The default). The input file specified in the Query/400 object is used.
file_name	The name of the file to be queried. This overrides the file name specified in the Query/400 object.

The possible library values are:

***LIBL**	All libraries in the user and system portions of the job's library list are searched until the first match is found.
***CURLIB**	The current library for the job is used to locate the database file. If no library is specified as the current library for the job, the QGPL library is used.
library-name	Specify the name of the library to be searched.

Query file member

This element has been deleted from the format-specific commands: use the QRYFILE parameter instead.

Specifies member in the file which the Query/400 query should process as its input file.

***FIRST**	(The default). The first member is used.
member_name	The name of the member to be used.

Output form

Specifies the form of output produced by the query. If no value was specified in the query and no value was entered on the command, or if a query name is not specified, *DETAIL is assumed.

The possible values are:

***RUNOPT**	If a query definition is used, the output form specified in the query definition is used when this query is run. If the query contains both detail fields and summary functions, Slipstream will run it in *COMBINED mode (see below).
***DETAIL**	The output form produced by the query is a report containing detail records and summary records if any exist.
***SUMMARY**	The output form produced by the query is a report containing summary records only.
***COMBINED**	Where a query contains both detail fields and summary functions, Slipstream will create a combined file containing both detail lines and summaries.

Include break level field?

When Query/400 produces summary output and that output is directed to an output file, the file created includes two additional columns: the break level, specifying the summary level to which the columns relate, and an overflow indicator, which flags field overflow if it occurs.

This element determines how Slipstream handles the first of those two additional fields (break level).

The possible values are:

***QRYDFNTXT**	The value of the break level field (a number representing the break level) is replaced by the corresponding level break text defined for the appropriate summary level in the Query/400 query.
***INCLUDE**	The value of the break level field is included as a column in the output. The column contains a number denoting the summary level to which the other columns relate.
***EXCLUDE**	The break level field is dropped from the output completely.
***LEVELBRK**	The break level field is replaced by a piece of text retrieved from message id SLP5001 in message file SL_MSGF, combined with the value of the break level field. For level 0 (grand totals) the text is derived from message SLP5002. You may modify this text if you wish but please note that these modifications will need to be repeated every time a Slipstream PTF or a new release is installed.
***SUBTOTAL**	The break level field is replaced by a piece of text retrieved from message id SLP5004 in message file SL_MSGF. This is useful if you just want the same word to appear each time, e.g. "Sub-total'. For level 0 (grand totals) the text is derived from message SLP5002. You may modify this text if you wish but please note that these modifications will need to be repeated every time a Slipstream PTF or a new release is installed. Note that the title of the column representing the break level is derived from message id SLP5003 in message file

SL_MSGF. You may modify this text if you wish but please note that these modifications will need to be repeated every time a Slipstream PTF or a new release is installed.

Include overflow indicator?

When Query/400 produces summary output and that output is directed to an output file, the file created includes two additional columns: the break level, specifying the summary level to which the columns relate, and an overflow indicator, which flags field overflow if it occurs.

This element determines how Slipstream handles the second of those two additional fields (overflow indicator).

The possible values are:

***EXCLUDE**	The overflow indicator field is dropped from the output completely.
***INCLUDE**	The value of the overflow indicator is included as a column in the output.

Example: CVTDBFXLS FROMFILE(*QRYDFN) …
QRYDFN(CUSTQRY *YES)

This example shows a Query/400 query call CUSTQRY being run to provide the input to the CVTDBFXLS command. If this command is run interactively, before the query is executed, you will be prompted with the Query/400 Record Selection screen to enable you to enter record selection criteria. The output from the query will be saved as an Excel file.

See the QRYSLT parameter for details of how that parameter relates to record selection performed with the query.

See the SORT parameter for details of how that parameter relates to record ordering performed with the query.

QRYFILE – Query file

Parameter	**QRYFILE**
Description	**The input file(s) to use with a Query/400 query**
Applies to commands:	**CVTDBFXLS, CVTDBFCSV, CVTDBFHTML, CVTDBFPDF, CVTDBFTXT, CVTDBFXML, CVTDBFPDF** **CVTDBFSTMF (deprecated)**
Dependent on:	**FROMFILE(*QRYDFN)**
Supports Slipstream variables	**No**
Migration notes	**The Query File and Query File Member elements of the QRYDFN parameter have been deleted from the new format-specific commands. If you have code which references this parameter, it will need to be modified when moving away from CVTDBFSTMF to the new format-specific commands if that code references these deleted elements or any later elements.** **The Query File and Query File Member elements are now replaced by the QRYFILE parameter (which previously provided an alternative means of specifying a query file and was the only way of specifying multiple query files).**

The **QRYFILE** (Query file) parameter only appears if **FROMFILE(*QRYDFN)** is specified, indicating that the input is to be the output created by running a Query/400 query. This parameter allows you to override the default names of the files used by your Query/400 query.

The default is the single value ***QRYDFN** which indicates that the default file names, librariy names and member names held in the Query/400 query object specified on the QRYDFN parameter will be used.

Alternatively you can specify the file, library and member name(s) of up to 32 files to be used instead of the files defined in the Query/400 object.

QMQRY – QM Query options

Parameter	**QMQRY**
Description	**QM query options when FROMFILE(*QMQRY) used**
Applies to commands:	**CVTDBFXLS, CVTDBFCSV, CVTDBFHTML, CVTDBFPDF, CVTDBFTXT, CVTDBFXML, CVTDBFPDF** **CVTDBFSTMF (deprecated)**
Dependent on:	**FROMFILE(*QMQRY)**
Supports Slipstream variables	**No**

The **QMQRY** (Query Management Query Options) parameter only appears if **FROMFILE(*QMQRY)** is specified, indicating that the input is to be the output created by running a Query Management (QM) query. This parameter allows you to indicate the name of the QM Query object to be run.

See the QRYSLT parameter for details of how that parameter relates to record selection performed with the query.

See the SORT parameter for details of how that parameter relates to record ordering performed with the query.

There are four elements to this parameter.

QM query object

The first element is the qualified name of the QM Query object.

query-name Specify the name of the QM Query object (object type *QMQRY) which should be run.

The possible library values are:

***LIBL** All libraries in the user and system portions of the job's library list are searched until the first match is found.

***CURLIB** The current library for the job is used to locate the database file. If no library is specified as the current library for the job, the QGPL library is used.

library-name Specify the name of the library to be searched.

Query Management report form

The next element is the qualified name of the QM Report Form object. ***Slipstream*** will retrieve column headings from the QM Report Form object you specify here. If you do not specify a QM Report Form object, headings will be taken from the output file created by running the QM Query.

***NONE**	(Default). No QM Form is specified and column headings are taken from the output file created when the QM Query is run.
***QMQRY**	The value specified on the Query management query prompt (QMQRY parameter) is used to locate the report form.
QM-form-name	Specify the name of the QM Form object (object type *QMFORM) from which headings will be retrieved.

The possible library values are:

***LIBL**	All libraries in the user and system portions of the job's library list are searched until the first match is found.
***CURLIB**	The current library for the job is used to locate the database file. If no library is specified as the current library for the job, the QGPL library is used.
library-name	Specify the name of the library to be searched.

Allow information from QRYDFN

This element specifies whether a query definition (*QRYDFN) object is used when no query management query (*QMQRY) object can be found using the specified object name.

The possible options are:

***NO**	An *QRYDFN object will not be used.
***YES**	An *QRYDFN object will be used if no *QMQRY object of the specified name is found.
***ONLY**	An *QRYDFN object will be used whether an *QMQRY object with the specified name exists or not.

Set variables

This element is a list of up to 50 variables referenced in the QM query and the value to be assigned to those variables when the query is run.

Values in this list appear in pairs of variables names and associated values. The variable name can be from 1 to 30 characters and the value from 1 to 55 characters in length. Enclose alphanumeric variable values in single quotes (‘) but leave numeric values without quotes.

FTP – FTP parameters

Parameter	**FTP**
Description	**FTP options to b used when TOSTMF(*FTP) specified**
Applies to commands:	**CVTDBFXLS, CVTDBFCSV, CVTDBFHTML, CVTDBFPDF, CVTDBFTXT, CVTDBFXML, CVTDBFPDF** **CVTDBFSTMF (deprecated)**
Dependent on:	**TOSTMF(*FTP)**
Supports Slipstream variables	**Yes**
Migration notes	**Three new parameter elements have been added in the middle of this parameter for the new format-specific commands. If you have code which references this parameter, it will need to be modified when moving away from CVTDBFSTMF to the new format-specific commands if that code references parameter elements after the insertion point for the new elements.**

The FTP parameter allows you to define parameters needed to transfer the output to an FTP server when TOSTMF(*FTP) is specified.

Secure FTP options have been added in this release and are not available on CVTDBFSTMF.

There are 10 elements to this parameter (only 7 for CVTDBFSTMF):

- **Remote system name/IP address**
- **Remote file path**
- **Port number**
- **Secure connection (not present on CVTDFBSTMF)**
- **Data protection (not present on CVTDFBSTMF)**
- **Remote user id**
- **Remote password**
- **Encrypted password supplied (not present on CVTDFBSTMF)**
- **Logging**
- **CCSID for log messages**

The default is the single value:

***NONE** Indicates that you do not intend to use FTP. Invalid if TOSTMF(*FTP) specified.

Remote system name/IP address

Specify the name of IP address of the system to which the data should be transmitted by FTP.

If you specify a name, the system i must be able to resolve that name to an IP address either by means of a DNS (Domain Name Server) or by looking up the name in the system Host Table.

Remote file path

Specify the full path where the output should be saved on the server. This should include both the name of the file to be created and the directory tree in which it should be saved.

Note that names on the server may be case-sensitive, especially if it is a UNIX system or similar, and may need to be enclosed in single quotes.

Port number

The port number to use,

Options are:

***FTP**	The default port for FTP (21) will be used.
***SECURE**	The default port for secure FTP (990) will be used.
Port_number	A valid port number between 1 and 65535.

Secure connection

The element does not exist for CVTDBFSTMF, which does not support FTP over SSL.

Specifies the type of security mechanism to be used for protecting information transferred on the FTP control connection (which includes the password used to authenticate the session with the FTP server). Transport Layer Security (TLS) and Secure Sockets Layer (SSL) are compatible protocols which use encryption to protect data from being viewed during transmission and verify that data loss or corruption does not occur.

Options are:

***NONE**	Slipstream client does not use encryption when connecting to the specified FTP server.
***IMPLICIT**	Slipstream immediately attempts to use TLS/SSL when connecting to the specified FTP server (without sending an AUTH subcommand to the server). If the server does not support implicit TLS/SSL on the specified port, or the TLS/SSL negotiation fails for any reason, the connection is closed.
***SSL**	After connecting to the specified FTP server, Slipstream sends an AUTH (authorization) subcommand requesting an SSL protected session. If the server does not support SSL, the connection is closed.
***TLS**	After connecting to the specified FTP server, Slipstream sends an AUTH (authorization) subcommand requesting a TLS protected session. If the server does not support TLS, the connection is closed.

Data protection

The element does not exist for CVTDBFSTMF, which does not support FTP over SSL.

Specifies the type of data protection to be used for information transferred on the FTP data connection. This connection is used to transfer file data and directory listings. The FTP protocol does not allow protection of the data connection if the control connection is not protected.

Note: The DTAPROT parameter controls the use of the PROT (protection) FTP server subcommand.

Options are:

***DFT**	If the SECCNN parameter specifies a protected control connection, *PRIVATE is used; otherwise, *CLEAR is used.
***PRIVATE**	Information sent on the FTP data connection is encrypted. If the SECCNN parameter specifies that the FTP control connection is not encrypted, *PRIVATE cannot be specified.
***CLEAR**	Information sent on the FTP data connection is not encrypted.

Remote user id

The user id to use when logging on. Names may be case sensitive and may need to be enclosed in single quotes.

Remote password

The password to use when logging on. Passwords may be case sensitive and may need to be enclosed in single quotes.

See the next element for details of how to supply this password in a scrambled form to avoid having to hold passwords in plain text form in source code.

When prompting the command, if you need to enlarge the size of this parameter element to allow specification of a hex string, enter an ampersand (&) then press return and OS/400 will increase the size the field.

If you need to enter a hex string, use the form X'0123456789ABCDEF' etc.

Encrypted password supplied

The element does not exist for CVTDBFSTMF.

Whether or not the password supplied on the previous element is supplied in the encrypted form returned by Slipstream's DSPENCPWD (Display Encrypted Password) command.

DSPENCPWD applies an encryption algorithm to a password and returns a scrambled version of that password to you. If you specify the scrambled password on the previous element, and specify *YES here, Slipstream will unscramble the password for you before sending it to the FTP server. The main purpose of this facility is to avoid the need to hold passwords in plain text form in source code.

Options are:

***NO**	The password supplied on the previous element is in plain text format and not scrambled.
***YES**	The password supplied on the previous is in the scrambled form returned by DSPENCPWD. It will be automatically unscrambled before being sent to the FTP server.

Logging

Whether to log messages relating to the FTP file transfer.

Options are:

***JOBLOG**	Messages will be logged to the joblog.
***NONE**	No messages will be logged.
Log_file_name	The name of the file to which messages should be logged. If a database file is to be used, specify the name in IFS format (e.g. '/QSYS.LIB/library_name.LIB/file_name.FILE/member_name.MBR')

CCSID for log messages

The CCSID in which messages should be written to the file.

Options are:

***CALC**	A CCSID will be calculated as follows. If the log file is a database file, the CCSID of the job will be used. Otherwise, the ISO ASCII equivalent of the CCSID of the job will be used.
CCSID	Specify the CCSID to use.

Example:

```
CVTDBFXLS    FROMFILE(SALES)…
             TOSTMF(*FTP)
             FTP(SalesSvr '/Sales/Sales.xls' *FTP 'BILL' 'soccer')
```

The sales file is converted to FTP and the output is sent directly to a server known to the system i as “SalesSvr” by FTP. The file will be saved in the “Sales” directory as “Sales.xls”. The port number will be 21. The connection will be established by logging on as BILL with the password “soccer”.

EMAIL – Email the output?

Parameter	**EMAIL**
Description	**Whether the file should be emailed after it is created/updated**
Applies to commands:	**CVTDBFXLS, CVTDBFCSV, CVTDBFHTML, CVTDBFPDF, CVTDBFTXT, CVTDBFXML, CVTDBFPDF** **CVTDBFSTMF (deprecated)**
Dependent on:	**None. Requires Communiqué or CoolSpools PLUS to be installed.**
Supports Slipstream variables	**No**

The **EMAIL** (Email the output) parameter enables you to specify that the stream file created by Slipstream should be emailed as an attachment.

Please note that this facility is only available if you have ariadne's ***Communiqué*** email product installed and licensed or on trial.

Options are:

***NO**	(The default). The output is not emailed automatically as part of running this command. You are still able to email the output separately, e.g. by running a subsequent email command such as SNDDST or Communiqué's SNDCMNMSG.
***YES**	The output from this command will be emailed as an attachment according to the information you specify on the other email-related parameters.

EMAILOPT – Email options

Parameter	**EMAILOPT**
Description	**Email-related options**
Applies to commands:	**CVTDBFXLS, CVTDBFCSV, CVTDBFHTML, CVTDBFPDF, CVTDBFTXT, CVTDBFXML, CVTDBFPDF** **CVTDBFSTMF (deprecated)**
Dependent on:	**EMAIL(*YES).** **Requires Communiqué or CoolSpools PLUS to be installed.**
Supports Slipstream variables	**Yes**

The **EMAILOPT** (Email options) parameter allows you to specify various options relating to the sending of the output from the command as an email attachment.

These options apply only to the sending of the output using ***Communiqué*** and have no effect on the sending of the output using subsequent calls to SNDDST and other email facilities.

The parameter consists of either 8 or 10 elements:

- Delete after sending?
- Subject
- Attach or embed? (present only on CVTDBFSTMF, CVTDBFTXT and CVTDBFHTML)
- Priority
- Confirm delivery
- Send multiple messages?
- Attachment name
- Zip attachment
- Zip file password
- Encrypted password supplied? (not available on CVTDBFSTMF)

Delete after sending

This option allows you to indicate whether the output from the command should be deleted as soon as it has been sent as an attachment.

Use this option with caution: if the email fails to arrive for whatever reason, you may lose your data.

Possible values are:

***NO**	(The default). The output is not deleted.
***YES**	Once the email has been created, and the stream file attached to it, the stream file is deleted. Please note that Slipstream can only tell if the email has been successfully created. It cannot tell if the email was successfully sent or delivered to its recipient.

Subject

This element allows you to define a subject line for the message. You can enter up to 50 characters of free-format text. When the email message is received, the text that you enter on this parameter element will appear in the subject line of the email.

Attach or embed?

This is where you specify the method by which the file is sent.

This option is only available on those commands which can generate HTML or text output (CVTDBFSTMF, CVTDBFHTML and CVTDBFTXT). The other commands generate binary output where the *EMBED option would be inappropriate.

Options are:

***ATTACH**	(Default) The file is sent as an attachment. It will appear as an attached file separate from the text of the email.
***EMBED**	The contents of the file are embedded in the text of the email and will follow the text of any message entered on the **EMAILMSG** parameter. Please note that your client email software is likely only to support the embedding of certain types of file, e.g. text and HTML.

Priority

The priority option controls whether the email message is flagged as a high-priority or low-priority in your email client software.

Values are:

***NORMAL**	(Default) The message is sent specifying normal priority. When the message arrives, the client email software will not mark it as high or low priority.
***HIGH**	High priority. When the message arrives, the client email software will mark it as high priority.
***LOW**	Low priority. When the message arrives, the client email software will mark it as low priority.

Confirm Delivery

This option controls whether confirmation of delivery is requested from the receiver of the email.

Values are:

***NO**	(Default) No confirmation of delivery if requested.
***YES**	The message is sent with an indication that you have requested that the recipient return confirmation of delivery. When the message is opened, if you have not switched off this feature, the client software will either send a confirmation message back to the sender of the email or ask you whether you wish to send such a confirmation.

Send Multiple Messages

This option determines whether, when multiple recipients are specified on the EMAILTO parameter, a single message is sent listing all of those recipients, or whether separate

messages are sent to each recipient. This controls whether or not each recipient is aware of how else has received the message.

Values are:

***NO**	A single message is sent to a list of recipients.
***YES**	Multiple messages are sent, one to each recipient.

Attachment name

The name to be given to the attachment in the email. The default is for the name to be the same as the name of the stream file being created, but can be overriden to something else if preferred.

Values are:

***TOSTMF**	The attachment name is the same as the name of the stream file being created.
attachment_name	Specify the name to be given to the attachment.

Zip attachment

Whether the attachment is sent inside a zip file or not.

Values are:

***NO**	The attachment is not zipped prior to being attached.
***YES**	The attachment is sent inside a zip file.

Zip file password

Where the attachment is sent inside a zip file, the optional password to encrypt that zip file.

Values are:

***NONE**	No zip file is used or the zip file is not encrypted.
zip_password	Specify the case-sensitive password for the zip file.

When prompting the command, if you need to enlarge the size of this parameter element to allow specification of a hex string, enter an ampersand (&) then press return and OS/400 will increase the size the field.

If you need to enter a hex string, use the form X'0123456789ABCDEF' etc.

Encrypted password supplied

The element does not exist for CVTDBFSTMF.

Whether or not the password supplied on the previous element is supplied in the encrypted form returned by Slipstream's DSPENCPWD (Display Encrypted Password) command.

DSPENCPWD applies an encryption algorithm to a password and returns a scrambled version of that password to you. If you specify the scrambled password on the previous element, and specify *YES here, Slipstream will unscramble the password for you before sending it to the FTP server. The main purpose of this facility is to avoid the need to hold passwords in plain text form in source code.

Options are:

***NO**	The password supplied on the previous element is in plain text format and not scrambled.

***YES**	The password supplied on the previous is in the scrambled form returned by DSPENCPWD. It will be automatically unscrambled before being sent to the FTP server.

EMAILFROM – Email sender information

Parameter	**EMAILFROM**
Description	**The email address and name from which the email will be sent**
Applies to commands:	**CVTDBFXLS, CVTDBFCSV, CVTDBFHTML, CVTDBFPDF, CVTDBFTXT, CVTDBFXML, CVTDBFPDF** **CVTDBFSTMF (deprecated)**
Dependent on:	**EMAIL(*YES).** **Requires Communiqué or CoolSpools PLUS to be installed.**
Supports Slipstream variables	**Yes**

The **EMAILFROM** (Email sender information) parameter allows you to specify the sender of the email and the email address to which a response should be sent.

The default value is ***CURRENT**, which means that ***Communiqué*** will try to retrieve the email address of the user sending the email from the System Distribution Directory. If no email address is defined for the user in the System Distribution Directory, you will need to enter the values you wish to use manually.

There are 2 elements to this parameter: **Email address** and **Name**.

Email address

This is where you enter the email address of the sender.

Note that while ***Communiqué*** will check that the email address that you enter conforms to the rules for valid email addresses, it is not possible to validate that the email address that you enter is correct or that any reply sent to the message will be deliverable.

For example, sales.ariadnesoftware.co.uk is not a valid email address (since it does not contain an @ sign), and ***Communiqué*** will reject it. However, sales@ariadnesoftware.org.uk is a valid email address and ***Communiqué*** will allow it, but it is not ariadne's correct email address (it should be sales@ariadnesoftware.co.uk) and any reply sent to this email address will not be received.

Name

If you would like you email message to display a sender's name rather than the sender email address when it is delivered, enter the name here.

The default value is ***NONE**, i.e. no name is provided and the email address will appear as the sender instead.

For example, if you specify:

EMAILFROM((Sales@ariadnesoftware.co.uk *NONE))

when the message is received, the **From:** attribute will be shown as:

From: Sales@ariadnesoftware.co.uk

However, if you specify:

EMAILFROM((Sales@ariadnesoftware.co.uk 'ariadne Sales'))

when the message is received, the **From:** attribute will be shown as:

From: ariadne Sales

EMAILTO – Email recipient(s)

Parameter	**EMAILTO**
Description	**The email address(es) and name(s) of the person(s) to whom the email will be sent**
Applies to commands:	**CVTDBFXLS, CVTDBFCSV, CVTDBFHTML, CVTDBFPDF, CVTDBFTXT, CVTDBFXML, CVTDBFPDF** **CVTDBFSTMF (deprecated)**
Dependent on:	**EMAIL(*YES).** **Requires Communiqué or CoolSpools PLUS to be installed.**
Supports Slipstream variables	**Yes**

The **EMAILTO** (Recipients) parameter allows you to specify the email addresses to which the email message should be sent.

You can define up to 32 recipients for the message on this command parameter. If you need to send the same email address to more than 32 recipients simultaneously, you can do this by defining an address list and specifying the address list name on this parameter.

The default is the single value ***SELECT:** Communiqué will prompt you to enter one or more email addresses to which the message should be sent, or you can select email addresses from email address directories. Please note that this feature is not available unless you have applied Communiqué PTF 1CM0053 or later.

There are 3 elements to this parameter: **Email address**, **Name** and **Type**.

Email address

This is where you enter the email address to which the message is to be sent.

Note that while ***Communiqué*** will check that the email address that you enter conforms to the rules for valid email addresses, it is not possible to validate that the email address that you enter is correct or that the message will be deliverable.

For example, sales.ariadnesoftware.co.uk is not a valid email address (since it does not contain an @ sign), and ***Communiqué*** will reject it. However, sales@ariadnesoftware.org.uk is a valid email address and ***Communiqué*** will allow it, but it is not ariadne's correct email address (it should be sales@ariadnesoftware.co.uk) and the message will not be received.

Name

If you would like your email message to display the recipient's name rather than the email address when it is delivered, enter the name here.

The default value is ***NONE**, i.e. no name is provided and the email address will appear as the recipient instead.

For example, if you specify:

EMAILTO((Sales@ariadnesoftware.co.uk *NONE))

when the message is received, the **To:** attribute will be shown as:

To: Sales@ariadnesoftware.co.uk

However, if you specify:

EMAILTO((Sales@ariadnesoftware.co.uk 'ariadne Sales'))

when the message is received, the **To:** attribute will be shown as:

To: ariadne Sales

Type

Specify the type of recipient here.

Options are:

***PRI**	(Default) Primary recipient.
***CC**	Carbon copy recipient. An *CC recipient receives a copy of the message, and is identified to the primary recipient, but is not the primary recipient.
***BCC**	Blind carbon copy recipient. An *BCC recipient receives a copy of the message, but is not identified to the primary recipient or *CC recipients.
***ADRL**	***Communiqué*** Address list. If you wish to send to an address list, this is the value that must be entered. Refer to the Communiqué manual for details of how to create, manage and use email address lists.

Example:

Sending to ariadne Sales as a primary recipient with a copy to ariadne Marketing:

```
CVTDBFXLS
EMAIL(*YES)
EMAILTO(    (Sales@ariadnesoftware.co.uk 'Sales' *PRI)
            (Marketing@ariadnesoftware.co.uk 'Marketing' *CC))
```

Example:

Sending to an email address list called "Sales":

EMAILTO((Sales *ADRL *ADRL))

EMAILMSG – Email message

Parameter	**EMAILMSG**
Description	**Defines an email message to be sent with the output file**
Applies to commands:	**CVTDBFXLS, CVTDBFCSV, CVTDBFHTML, CVTDBFPDF, CVTDBFTXT, CVTDBFXML, CVTDBFPDF** **CVTDBFSTMF (deprecated)**
Dependent on:	**EMAIL(*YES).** **Requires Communiqué or CoolSpools PLUS to be installed.**
Supports Slipstream variables	**Yes**

The **EMAILMSG** (Message) parameter allows you to enter the text of an email message directly on the command line.

Up to 512 characters of free-format text can be entered here.

The message can be sent in either plain text, HTML or alternative plain text/HTML formats.

There are 3 elements to this parameter:

- Message text
- Message format.
- Text or path name specified?

Message text

This element plays two roles, depending on the value of the third element (Text or path name specified?):

- If “Text or path name specified?” is *MSG, this element holds a free-form text message
- If “Text or path name specified?” is *STMF, this element holds the path to a file containing the text of a message.
- If If “Text or path name specified?” is *MSG, You can enter the text of a short message to accompany the email and its attachment here. Up to 512 characters can be entered. When received, the message will be displayed exactly as it is entered, with the following exceptions:
- If you want to force a line break, enter <br>. Even if the message is sent in plain text format, this HTML control will be interpreted and converted to a hard line break (carriage return-line feed sequence).
- Other HTML controls may be entered, but will only be interpreted as HTML controls if the message is sent and delivered in HTML format.
- When received, the message will be displayed exactly as it is entered, with the following exceptions:

Message format

This is where you specify the format in which the message is sent.

Options are:

***BOTH**	(Default) The message is sent in alternative plain text/HTML format. This means that two copies of the message text will be sent: a plain text copy and an HTML copy. If the email client software used to receive the message can handle HTML messages, the HTML copy will be used, otherwise the plain text copy will be used.
***TEXT**	The message is sent in plain text format. The only HTML control which is interpreted is , which ***Communiqué*** will convert to a hard line break.
***HTML**	The message is sent in HTML format. You can include HTML formatting (e.g. or <u> </u> to control bold text and underlining). ***Communiqué*** will take the text that you enter and wrap it with some basic HTML header and footer controls (<HTML> <HEAD> <BODY>). These controls should not therefore be included in the text of the message.

Text or path name specified?

Controls the interpretation of the first element of this parameter. See above.

Options are:

***MSG**	The first element specified is a free-format message text.
***STMF**	The first element specified is the path name of a stream file containing the text of the message to be sent.

Example:

```
CVTDBFXLS...
EMAIL(*YES)
EMAILMSG('Here''s a message <br>with<br>line <br>breaks.' *TEXT)
```

When this message is received, it will show as:

```
Here's a message
with
line
breaks.
```

Example:

```
CVTDBFXLS...
EMAIL(*YES)
EMAILMSG( 'Here''s a message with HTML controls.<br>
          <b>This line is in bold, </b><br>
          <u>While this line is underlined.</u>')
```

When this message is received, it will show as:

Here's a message with HTML controls.
This line is in bold,
While this line is underlined.

RCDFMT – Record format

Parameter	**RCDFMT**
Description	**Defines the record format(s) to be converted**
Applies to commands:	**CVTDBFXLS, CVTDBFCSV, CVTDBFHTML, CVTDBFPDF, CVTDBFTXT, CVTDBFXML, CVTDBFPDF** **CVTDBFSTMF (deprecated)**
Dependent on:	**None**
Supports Slipstream variables	**No**

The **RCDFMT** (Record Formats) parameter enables you to select which record formats from the input file are included in the conversion process.

There is one single option:

***ONLY**	(Default) The input file contains only a single record format. Use this option for all database files other than logical files which contain more than one record format.

Alternatively, specify a list of from one to 20 record format names from the input file that should be included in the conversion process.

If you prompt the **CVTDBFXLS** command, and specify *YES for the “Select records and fields” component of the **FROMFILE** parameter, ***Slipstream*** will display a list of up to 20 record format names from the input file for you to select from.

INCLFLD – Include fields

Parameter	**INCLFLD**
Description	**Lists fields to be included in the output**
Applies to commands:	**CVTDBFXLS, CVTDBFCSV, CVTDBFHTML, CVTDBFPDF, CVTDBFTXT, CVTDBFXML, CVTDBFPDF** **CVTDBFSTMF (deprecated)**
Dependent on:	**None**
Supports Slipstream variables	**No**

The **INCLFLD** (Include Fields) parameter enables you to select which fields from the input file will be included in the output stream file, and the sequence in which they will be presented.

There is one single option:

***ALL** (Default) All fields from the input file are included and they occur in the stream in the order in which they occur in the input file. However, any fields specified on the **EXCLFLD** (Exclude Fields) parameter (see below) will be omitted.

Alternatively, specify a list of from one to 300 qualified field names from the input file that should be included in the conversion process. These fields will be presented in the stream file in the order in which they are listed on this parameter.

If you prompt the **CVTDBFXLS** command, and specify *YES for the “Select records and fields” component of the **FROMFILE** parameter, ***Slipstream*** will display a list of up to 300 qualified fields names from the input file for you to select from.

Each qualified field name consists of the field name and a qualifying record format name. This allows ***Slipstream*** to distinguish between fields of the same name in different record formats. If the input file contains only a single record format name, the special value ***ONLY** (the default) can be specified for the record format name, indicating that the field is to be taken from the single record format in the file.

EXCLFLD – Exclude fields

Parameter	**EXCLFLD**
Description	**Lists fields to be excluded from the output**
Applies to commands:	**CVTDBFXLS, CVTDBFCSV, CVTDBFHTML, CVTDBFPDF, CVTDBFTXT, CVTDBFXML, CVTDBFPDF** **CVTDBFSTMF (deprecated)**
Dependent on:	**None**
Supports Slipstream variables	**No**

The **EXCLFLD** (Exclude Fields) parameter enables you to specify fields which should not be included in the output stream file. If your input file contains a large number of fields, and you wish to exclude only a small number of fields, it may be more convenient to specify the few fields to be excluded by name rather than list the large number of fields to be included.

There is one single option:

***NONE**	(Default) No fields from the input file are excluded. However, if a value other than ***ALL** is input for the **INCLFLD** (Include Fields) parameter, only the fields listed there will be included in the output stream file.

Alternatively, specify a list of from one to 300 qualified field names from the input file that should be excluded in the conversion process. These fields will be omitted from the data in the stream file.

Each qualified field name consists of the field name and a qualifying record format name. This allows ***Slipstream*** to distinguish between fields of the same name in different record formats. If the input file contains only a single record format name, the special value ***ONLY** (the default) can be specified for the record format name, indicating that the field is to be taken from the single record format in the file.

EXCEL – Excel options

Parameter	**EXCEL**
Description	**Options specific to Excel output**
Applies to commands:	**CVTDBFXLS** **CVTDBFSTMF (deprecated)**
Dependent on:	**TOFMT(*XLS) (CVTDBFSTMF only)**
Supports Slipstream variables	**Yes (all elements that accept a free-form text value)**
Migration notes	**The EXCEL parameter of CVTDBFXLS has been considerably simplified compared with the equivalent parameter of CVTDBFSTMF through the creation of a separate XLSPRPRTY parameter where Excel file properties are now defined.** **By contrast, some new elements are only available on CVTDBFXLS while others which has been superseded are available only on CVTDBFSTMF. See below for details.**

When using CVTDBFSTMF, the **EXCEL** (Excel options) parameter only appears if **TOFMT(*XLS)** is selected. This parameter allows you to control many aspects of the process of creating an Excel format file from your database file.

Excel file format version

This option allows you to select the version of the Excel (BIFF) file format that will be implemented when the Excel file is created.

***XLS**	(Default) BIFF 8 format is used. BIFF 8 is the format used by versions of Excel from Excel 97 onwards. If your PC program supports this format, then it is recommended format that this is the option you should choose, as BIFF 8 files are normally more compact than BIFF 5 files.
***XLSX**	Excel 2007 Open Office XML format (.xlsx file).
***BIFF8**	The same as *XLS, provided for backwards-compatibility.

Note that support for BIFF5 format (Excel 95) is now withdrawn in this release.

Note the following limits imposed by Excel (not Slipstream):

Attribute	BIFF 8 Maximum	Open Office XML Maximum
Maximum rows in a worksheet	65,536	1,048,576
Maximum columns in a worksheet	256	16,384

If the number of records in the input file exceeds the maximum number of rows per worksheet for the format being implemented, ***Slipstream*** will create additional worksheets for the overflow.
If the number of fields in the record being converted exceeds the maximum number of columns per worksheet for the format being implemented, ***Slipstream*** will drop fields beyond the maximum.

Default edit code

Not available on CVTDBFXLS. The new User-defined Styles (DFNSTYLES) and FLDSTYLES (Field Styles) parameters, available on the CVTDBFXLS command, provide greater control over the formatting of data and use of this feature is now deprecated.

This option controls the format in which numeric information in the input file is presented in the Excel file.

Where numeric fields in the input file have DDS edit codes or edit words defined for them, ***Slipstream*** will convert the edit code or edit word to an Excel custom format for the corresponding column in the spreadsheet. This means that the data in the column will be displayed in a format which reflects the edit code or edit word of the original database field.

Where numeric fields in the input file do not have a DDS edit code or edit word associated with them, ***Slipstream*** will format the data in the spreadsheet according to the value you enter on this parameter instead.

The value you enter must be ***NONE** (the default) to indicate that you do not wish to have numeric data edited in this way, or a valid system i edit code from the following table.

Edit Codes	Commas[1] Displayed	Decimal Points[1] Displayed	Sign Displayed When Negative Value	Blank Value of QDECFMT System Value	I Value of QDECFMT System Value	J Value of QDECFMT System Value	Leading Zero Suppressed
1	Yes	Yes	No sign	.00 or 0	,00 or 0	0,00 or 0	Yes
2	Yes	Yes	No sign	Blanks	Blanks	Blanks	Yes
3		Yes	No sign	.00 or 0	,00 or 0	0,00 or 0	Yes
4		Yes	No sign	Blanks	Blanks	Blanks	Yes
A	Yes	Yes	CR	.00 or 0	,00 or 0	0,00 or 0	Yes
B	Yes	Yes	CR	Blanks	Blanks	Blanks	Yes
C		Yes	CR	.00 or 0	,00 or 0	0,00 or 0	Yes
D		Yes	CR	Blanks	Blanks	Blanks	Yes
J	Yes	Yes	-(Minus)	.00 or 0	,00 or 0	0,00 or 0	Yes
K	Yes	Yes	-(Minus)	Blanks	Blanks	Blanks	Yes
L		Yes	-(Minus)	.00 or 0	,00 or 0	0,00 or 0	Yes
M		Yes	-(Minus)	Blanks	Blanks	Blanks	Yes
N	Yes	Yes	-(Minus)	.00 or 0	,00 or 0	0,00 or 0	Yes
O	Yes	Yes	-(Minus)	Blanks	Blanks	Blanks	Yes
P		Yes	-(Minus)	.00 or 0	,00 or 0	0,00 or 0	Yes
Q		Yes	-(Minus)	Blanks	Blanks	Blanks	Yes
W[2]							Yes
Y[3]							Yes
Z[4]							Yes

Notes:

1. The QDECFMT system value determines the decimal point character (period in U.S. usage), the character used to separate groups of three digits (comma in U.S. usage), and the type of zero suppression (depending on comma and period placement).
2. The W edit code suppresses the farthest left zero of a date field that is five digits long. It also suppresses the three farthest left zeros of a field that is six to eight digits long. The W edit code also inserts slashes (/) between the month, day, and year according to the following pattern:

   nn/nnn

   nnnn/nn

   nnnn/nnn

   nnnn/nn/nn

3. The Y edit code suppresses the farthest left zero of a date field that is three to six digits long or eight digits long. It also suppresses the two farthest left zeros of a field that is seven positions long. The Y edit code also inserts slashes (/) between the month, day, and year according to the following pattern:

   nn/n

   nn/nn

   nn/nn/n

   nn/nn/nn

   nnn/nn/nn

   nn/nn/nnnn

   If the DATE keyword is specified with EDTCDE(Y), the separator character used is the job attribute, DATSEP at run time. The slash (/) is the default DATSEP.

4. The Z edit code removes the sign (plus and minus) from a numeric field. The sign of the units position is changed to a hexadecimal F before the field is written.

Floating currency symbol

Not available on CVTDBFXLS. Use the new User-defined Styles (DFNSTYLES) and FLDSTYLES (Field Styles) parameters instead as these provide greater control over the formatting of data and use of this feature is now deprecated.

This option controls whether, when a default edit code has been specified on the previous option, a currency symbol (corresponding to the system value QCURSYM) is displayed immediately to the left of the lost-most significant digit.

***NO**	(Default). No floating currency symbol appears.
***YES**	Numeric data edited using the default edit code (described above) is preceded by the floating currency symbol. The currency symbol is taken from the QCURSYM system value.

Date format

This option controls the default format in which dates from the input file are presented in the Excel file.

Excel represents dates as a count of days. ***Slipstream*** will convert date fields in the input database file to an integer cell containing the appropriate day count and will apply formatting to display the value as a date. This makes it easy for you to apply formulas to the date subsequently.

In the main, the format in which date information is displayed in the Excel spreadsheet is determined by the regional settings on your PC. This parameter provides some options to vary that formatting slightly.

You can specify one of the following single values for this option:

***MM**	(Default) Months are displayed as 2-digit numeric values, e.g. 01=January.
***MMM**	Months are displayed as 3-character values, e.g. JAN=January.
***LABEL**	Dates are treated in the same way as character fields and output as an alphanumeric label cell rather than as an integer cell with date formatting.

Time format

This option controls the format in which time data from the input file is presented in the Excel file. Please note that this only applies to actual time fields in the database file: numeric and alphanumeric fields which contain times cannot be identified as times by ***Slipstream***.

Excel represents times as a fraction of a day, e.g. 12 noon = 0.5. ***Slipstream*** will convert time fields in the input database file to a cell containing a floating point value representing a day fraction of this type and will apply formatting to display the value as a time. This parameter controls the type of formatting that is applied.

You can specify one of the following single values for this option:

***HMS**	(Default) Times are displayed using the 24-hour clock including seconds, e.g. 3:30 in the afternoon is displayed something like 15:30:00.
***HM**	Times are displayed using the 24-hour clock excluding seconds, e.g. 3:30 in the afternoon is displayed something like 15:30.
***HMSAM**	Times are displayed using the AM/PM format including seconds, e.g. 3:30 in the afternoon is displayed something like 3:30:00 PM.
***HMAM**	Times are displayed using the AM/PM format excluding seconds, e.g. 3:30 in the afternoon is displayed something like 3:30 PM.

The actual format of the time displayed may differ from these examples (in particular the separator character may vary) as the formatting is also influenced by the regional options on the PC.

Worksheet name

This option allows you to specify the name given to worksheets in the Excel file.

Options are:

***DFT**	(Default) The default worksheet names are used. The default worksheet name is stored in the message text of message SLP5000 in the Slipstream message file SL_MSGF. It is supplied as “Sheet1” (the standard Excel English-language worksheet name) when Slipstream is first installed. However, you can change the text of this message if you wish to modify the default worksheet name. Note that you will need to repeat this change each time a PTF or new version of Slipstream is installed.
‘Sheet name’	Enter a valid Excel worksheet name. Excel worksheets names are from 1-31 characters in length and can contain any characters except: colon (:), backslash (\), forward slash (/), question mark (?), asterisk (*), left-hand square bracket ([] and right-hand square bracket (]).

If the data will not fit into a single worksheet, ***Slipstream*** will create additional worksheets up to a maximum of 36. The names of the second worksheet and subsequent worksheets are based on the value specified on this parameter according to the following rules:

1. If the name specified on this parameter, or the text retrieved from message id SLP5000 in message file SL_MSGF if *DFT is specified, ends in a ‘1’ (e.g. English “Sheet1”, or German “Blatt1” or Dutch “Blad1”), this suffix is replaced by 2, 3, 4… (e.g. “Sheet2”, ”Sheet3”..., “Blatt2”, Blatt3”..., “Blad2”, Blad3”...).
2. If the name does not end in a 1, a numeric suffix is appended to the name. For example, if the name is “Invoices”, subsequent worksheets will be called “Invoices2”, “Invoices3”...

Column width option

Determines the way in which Slipstream sets the column width.

***FIELDSIZE**	(Default) The column width is calculated based on the maximum available size of the corresponding database/query field. For example, if the column is derived from an address line which can be up to 50 characters long, the column will be wide enough to hold 50 characters.
***AUTOFIT**	The column width is calculated based on the maximum used size of the corresponding database/query field. For example, if the column is derived from an address line which can be up to 50 characters long, but the longest address line in the input data is 20 characters, the column will be wide enough to hold 20 characters.

Note:

The way in which Excel calculates column widths is complex. More specifically, it is dependent on the font metrics of the default font for the workbook. Since Slipstream runs on the iSeries and cannot therefore readily access those font metrics, when you specify a font other than one of Slipstream’s “well known” fonts (Courier New, Arial, Times Roman), or

where you are adding to an existing spreadsheet that uses a font other than the “well known” fonts, Slipstream cannot guarantee to calculate columns widths precisely.

Maximum rows per worksheet

The maximum number of rows that will be written to a worksheet before starting a new worksheet.

Options are:

***XLSVER**	(Default) The maximum is dictated by the version of Excel being output (65536 for *XLS and 1048576 for *XLSX)
1-65536	Specify the maximum. When this number of rows has been written to a worksheet, Excel will start a new overflow worksheet.

Hide unused columns

Not available from CVTDBFSTMF.

Whether unused columns are hidden or not.

Options are:

***NO**	(Default) Unused columns are not hidden and will appear as empty columns to the right of the last used column of data.
***YES**	Empty columns will be hidden and the last used column will be the last visible column in the worksheet.

Hide unused rows

Not available from CVTDBFSTMF.

Whether unused rows are hidden or not.

Options are:

***NO**	(Default) Unused rows are not hidden and will appear as empty rows to the right of the last used row of data.
***YES**	Empty rows will be hidden and the last used row will be the last visible row in the worksheet.

Title

See XLSPRPRTY parameter.

Subject

See XLSPRPRTY parameter.

Author

See XLSPRPRTY parameter.

Manager

See XLSPRPRTY parameter.

Company

See XLSPRPRTY parameter.

Category

See XLSPRPRTY parameter.

Keywords

See XLSPRPRTY parameter.

Comments

See XLSPRPRTY parameter.

Document content status

See XLSPRPRTY parameter.

Font Name

CVTDBFSTMF only. Use of this option is now deprecated. Use the DFNSTYLES and FLDSTYLES parameters to define named styles instead.

Specifies the font to be used in the Excel file.

***ARIAL**	(Default) The Arial font is used.
***COURIER**	Courier New.
***TIMES**	Times New Roman
***CALIBRI**	Calibri
Font_name	Specify the name of the font to be used. This must match a font installed on the PC which opens the file, otherwise Excel will substitute a different font.

Font Size

CVTDBFSTMF only. Use of this option is now deprecated. Use the DFNSTYLES and FLDSTYLES parameters to define named styles instead.

Specify the size of the font to be used in the Excel file, measured in points.

10	(Default)A 10-point font is used.
Font_size	Specify the size of the font in points.

XLSPRPRTY – Document properties

Parameter	**XLSPRPRTY**
Description	**Specifies document properties for Excel files.**
Applies to commands:	**CVTDBFXLS** **CVTDBFSTMF (deprecated)**
Dependent on:	**TOFMT(*XLS) (CVTDBFSTMF only)**
Supports Slipstream variables	**Yes**
Migration notes	**The EXCEL parameter of CVTDBFXLS has been considerably simplified compared with the equivalent parameter of CVTDBFSTMF through the creation of this separate XLSPRPRTY parameter where Excel file properties are now defined.**

This parameter allows you to define file properties for documentation and audit purposes.

The information defined here appears in Excel 2007 when you select:

Office button -> Prepare -> Properties

Title

***NONE**	(Default) The file will have no title.
Title_text	Up to 32 characters of title text.

Subject

***NONE**	(Default) The file will have no subject.
Subject_text	Up to 32 characters of subject text.

Author

A number of special values are available to help you use this field to document the origin of the file.

***NONE**	(Default) The file will have no title.
***USRPRF**	The user id of the user that created the file, e.g. SALESUSER.
***JOB**	The name of the job that created the file, e.g. SALESJOB,
***QUALJOB**	The qualified name of the job that created the file, e.g. 123456/SALESUSER/SALESJOB.
Author_text	Up to 32 characters of author text.

Manager

***NONE**	(Default) The file will have no manager.
Manager_text	Up to 32 characters of manager text.

Company

***NONE**	(Default) The file will have no company.

Company_text	Up to 32 characters of company text.

Category

***NONE**	(Default) The file will have no category.
Category_text	Up to 32 characters of category text.

Keywords

***NONE**	(Default) The file will have no keywords.
Keywords_text	Up to 128 characters of keywords text.

Comments

***NONE**	(Default) The file will have no comments.
Comments_text	Up to 256 characters of comments text.

Document content status

***NONE**	(Default) The file will have no document content status
Status_text	Up to 32 characters of text describing the status of the document content (e.g. “Draft”, “Final”, “Approved” etc.)

XLSPROTECT – Excel worksheet protection

Parameter	**XLSPROTECT**
Description	**Excel worksheet protection**
Applies to commands:	**CVTDBFXLS** **CVTDBFSTMF (deprecated)**
Dependent on:	**TOFMT(*XLS) (CVTDBFSTMF only)**
Supports Slipstream variables	**Yes**
Migration notes	**Note that CVTDBFXLS has an extra element compared with CVTDBFSTMF. Note also that columns can now be optionally left unlocked when a worksheet is protected by using a named style with the locking attribute set appropriately.**

This parameter allows you to specify options related to worksheet protection.

The default is the single option ***NO**, which indicates that no protection will be applied to the worksheet.

Note that if using STMFOPT(*ADD), this parameter only affects new worksheets added to the file and does not modify the protection status of existing worksheets in the file.

Elements are as follows.

Protect worksheet

Specify ***YES** to activate worksheet protection. When the file is opened, the worksheet will be protected. Cells to which a style has been supplied that indicates that cells should be locked will not be modifiable, except as permitted by the options specified below.

Worksheet protection password

Specifies the password that must be entered to unprotect the worksheet.

***NONE**	(Default)No password is required. The worksheet can be unprotected simply by taking the appropriate menu option.
Password	Specify the password that must be entered in order to unprotect the worksheet. This is case-sensitive and a maximum of 32 characters in length.

Encrypted password supplied

The element does not exist for CVTDBFSTMF.

Whether or not the password supplied on the previous element is supplied in the encrypted form returned by Slipstream's DSPENCPWD (Display Encrypted Password) command.

DSPENCPWD applies an encryption algorithm to a password and returns a scrambled version of that password to you. If you specify the scrambled password on the previous element, and specify *YES here, Slipstream will unscramble the password for you before

sending it to the FTP server. The main purpose of this facility is to avoid the need to hold passwords in plain text form in source code.

Options are:

***NO**	The password supplied on the previous element is in plain text format and not scrambled.
***YES**	The password supplied on the previous is in the scrambled form returned by DSPENCPWD. It will be automatically unscrambled before being sent to the FTP server.

Allow actions

Defines the actions that can be applied to locked items on a protected worksheet.

Single options are:

***DFT**	(Default) The actions allowed by Excel by default when a worksheet is protected are permitted. Both locked and unlocked cells may be selected, and objects and scenarios may be edited.
***NONE**	No actions are permitted on locked cells.

Alternatively, specify the actions to be permitted from the following list:

***DLTCOLS**	Deletion of columns
***DLTROWS**	Deletion of rows
***AUTOFILTER**	Applying autofilters
***EDTOBJ**	Editing objects
***EDTSCN**	Editing scenarios
***FMTCELLS**	Changing the formatting of cells
***FMTCOLS**	Changing the formatting of columns
***FMTROWS**	Changing the formatting of rows
***INSCOLS**	Inserting columns
***INSROWS**	Inserting rows
***INSLINKS**	Inserting hyperlinks
***PIVOTTABLE**	Applying pivot tables
***SLTUNLOCKED**	Selecting unlocked cells
***SLTLOCKED**	Selecting locked cells
***SORT**	Sorting rows

XLSPRINT – Excel print setup

Parameter	**XLSPRINT**
Description	**Specifies Excel print options**
Applies to commands:	**CVTDBFXLS** **CVTDBFSTMF (deprecated)**
Dependent on:	**TOFMT(*XLS) (CVTDBFSTMF only)**
Supports Slipstream variables	**Yes.** **Excel placeholders can also be specified on the header and footer text options.**

Specifies print options for Excel spreadsheets.

Single values:

***DFT** — No print options are specified in the Excel file. Excel defaults will be used.

Other values:

Scaling

How the data is enlarged or reduced when you print so that it fits the required number of pages.

Specify *FIT and a number of pages wide and tall on the XLSFITPAGES

parameter to fit the data to the required number of pages.

Specify *ADJUST and a percentage on the XLSADJUST parameter to

scale the data by that percentage.

Options are:

***FIT** — Fit the data to a number of pages wide and a number of pages tall. The number of pages wide and tall are specified on the dependent parameter XLSFIT.

***ADJUST** — Adjust the data by applying a percentage scaling. The percentage by which the data is scaled is specified on the dependent parameter XLSADJUST.

Page size

The paper size.

Options are:

***CNTRYID** — The paper size is determined by the country id of the job. If this is US (USA) or CA (Canada), letter paper is selected, otherwise A4 paper is selected.

***A3** — A3 (420 x 297 mm).

***A4** — A4 (297 x 210 mm).

***A5** — A5 (210 x 148 mm).

***B4**	B4 (364 x 257 mm).
***B5**	B5 (257 x 182 mm).
***LETTER**	Letter (11.5 x 8 inches).
***LEGAL**	Legal (14 x 8.5 inches).
***EXEC**	Executive (10.5 x 7.25 inches).
***LEDGER**	Ledger (17 x 11 inches).

Orientation

The page orientation.

Options are:

***LANDSCAPE**	Landscape mode.
***PORTRAIT**	Portrait mode.

Print gridlines

Whether gridlines should be printed or not.

Options are:

***NO**	Gridlines are not printed.
***YES**	Gridlines are printed.

Printer header row on each page

Whether any header row should be printed on each page.

Options are:

***NO**	The header row, if requested on the HEADER parameter, is printed only on the first page.
***YES**	The header row, if requested on the HEADER parameter, is printed on each page.

Unit of measure

The unit of measure in which margins are defined (see below)

Options are:

***INCH**	Inches
***MM**	Millimeters
***CM**	Centimeters

Left margin

The left page margin measured in the units specified (see Unit of Measure above).

Right margin

The right page margin measured in the units specified (see Unit of Measure above).

Top margin

The top page margin measured in the units specified (see Unit of Measure above).

Bottom margin

The bottom page margin measured in the units specified (see Unit of Measure above).

Page header left section

The text to appear in the left section of the page header.

Slipstream variables and Excel placeholders are supported on this parameter.

Page header center section

The text to appear in the center section of the page header.

Slipstream variables and Excel placeholders are supported on this parameter.

Page header right section

The text to appear in the right section of the page header.

Slipstream variables and Excel placeholders are supported on this parameter.

Page footer left section

The text to appear in the left section of the page footer.

Slipstream variables and Excel placeholders are supported on this parameter.

Page footer center section

The text to appear in the center section of the page footer.

Slipstream variables and Excel placeholders are supported on this parameter.

Page footer right section

The text to appear in the right section of the page footer.

Slipstream variables and Excel placeholders are supported on this parameter.

XLSADJUST – Adjust pages to

Parameter	**XLSADJUST**
Description	**Specifies the percentage sacling when XLSPRINT(*ADJUST...) is requested.**
Applies to commands:	**CVTDBFXLS** **CVTDBFSTMF (deprecated)**
Dependent on:	**XLSPRINT(*ADJUST)** **TOFMT(*XLS) (CVTDBFSTMF only)**
Supports Slipstream variables	**No**

Specifies the percentage scaling when XLSPRINT(*ADJUST...) is used.

Options are:

100	Scale by 100% (no change).
0-400	Specify the percentage scaling.

XLSFIT – Fit pages to

Parameter	**XLSFIT**
Description	**Specifies the number of pages to fit the output to when XLSPRINT(*FITPAGE...) is requested.**
Applies to commands:	**CVTDBFXLS** **CVTDBFSTMF (deprecated)**
Dependent on:	**XLSPRINT(*FIT)** **TOFMT(*XLS) (CVTDBFSTMF only)**
Supports Slipstream variables	**No**

Specifies the number of pages to which the output is fitted when XLSPRINT(*FIT...) is used.

There are two elements:

The number of pages wide (horizontal).

Options are:

***AUTO**	Excel will calculate the number of pages required automatically.

0-65535	Specify the number of pages to which the data should be fitted horizontally.

The number of pages tall (vertical).

Options are:

***AUTO**	Excel will calculate the number of pages required automatically.
0-65535	Specify the number of pages to which the data should be fitted vertically.

CSV – CSV options

Parameter	**CSV**
Description	**Specifies options for delimited ASCII text output (typically comma-separated variable but also tab-separated etc.)**
Applies to commands:	**CVTDBFCSV** **CVTDBFSTMF (deprecated)**
Dependent on:	**TOFMT(*CSV) (CVTDBFSTMF only)**
Supports Slipstream variables	**No**
Migration notes	**The format of this parameter has been altered somewhat between the CVTDBFSTMF and CVTDBFCSV commands. Specifically, CVTDBFSTMF allows the use of a single value *DFT which is not available on CVTDBFCSV. Also, on the CSV parameter of the CVTDBFSTMF command, the date format and date separator formed a related pair for which a single value could be specified, as did the time format and time separator. This tended to make use of this parameter excessively complicated. The CSV parameter of the CVTDBFCSV command has been re-designed to improve ease of use, but code changes may be necessary when migrating from CVTDBFSTMF ...TOFMT(*CSV) to use of the CVTDBFCSV command.**

The **CSV** (CSV options) parameter only appears if **TOFMT(*CSV)** is selected. This parameter allows you to control several aspects of the process of creating a delimited ASCII file from your database file.

Field delimiter

This option allows you to define the character that separates fields in the delimited ASCII file that is to be created.

Either type the character to be used, or select one of the special values:

***COMMA**	(Default) A comma (,) is used
***TAB**	A tab (x'09') is used
***BLANK**	A blank (x'20') is used
***SEMICOLON**	A semicolon (;) is used.

Example: ***CVTDBFCSV FROMFILE(SALESSTATS)...***
CSV(*COMMA *DBLQUOTE *CRLF)

The Sales Stats file is converted to a delimited file in CSV (Comma-separated variable format). Fields are separated by commas. Alphanumeric data is enclosed in double quotes. Records are terminated by a carriage return/line feed pair.

String delimiter

This element allows you to define the character that encloses string (alphanumeric) data in the delimited ASCII file that is to be created.

Either type the character to be used, or select one of the special values:

***DBLQUOTE**	A double quote (“) is used
***SGLQUOTE**	A single quote (‘) is used
***NONE**	No delimiter is used. Alphanumeric data is not enclosed by any special character.

Record delimiter

This element allows you to specify the characters to be used to indicate the end of a record in the CSV file.

Options are:

***CRLF**	Carriage return and line feed. Both a carriage return (x’0D’) and a line feed (x’0A’) character are used.
***CR**	Just a carriage return (x’0D’) is used.
***LF**	Just a line feed (x’0A’) is used.

Date format

This option controls the format in which date information in the database file is presented in the stream file. Please note that this only applies to actual date fields in the database file: numeric and alphanumeric fields which contain date information cannot be identified as dates by ***Slipstream***.

***DBF**	(Default). The format of the date is determined by the date format specified for the date field in the database file (DDS DATFMT and DATSEP keywords).
***EXCEL**	The date is converted to Excel format, i.e. a numeric value representing a day count. This format is ideal if you are going to load the CSV file into Excel or a similar application. Excel will recognize the data as a date and allow date functions and editing to be applied to it.
***ISO**	ISO format (YYYY-MM-DD) is used.
***EUR**	European format (DD.MM.YYYY) is used.
***USA**	USA format (MM/DD/YYYY) is used.
***JIS**	Japanese Industrial Standard (YYYY-MM-DD) is used.
***SYSVAL**	The date format defined in system value QDATFMT is used.
***JOB**	The date format defined in the job attributes is used.
***DMY**	DDMMYY format.
***DMYY**	DDMMYYYY format.
***MDYY**	MMDDYYYY format.
***YYMD**	YYYYMMDD format.
***CDMY**	CDDMMYY format. The C indicates the century (0 = 20th, 1=21st)
***CMDY**	CMMDDYY format. The C indicates the century (0 = 20th, 1=21st)
***CYMD**	CYYMMDD format. The C indicates the century (0 = 20th, 1=21st)
***JUL**	YYDDD format
***LONGJUL**	YYYYDDD format

Date separator

***JOB**	The date separator character defined in the job attributes is used.
***NONE**	No date separator character is used.
***SLASH**	A forward slash (oblique or solidus) / is used.
***HYPHEN**	A hyphen (dash) – is used
***PERIOD**	A period (full stop) . is used.
***COMMA**	A comma , is used.
***COLON**	A colon : is used.
***BLANK**	A blank is used.
separator_char	Type the separator character to be used to separate the day, month and year portions of the date.

Time format

This option controls the format in which time information in the database file is presented in the stream file. Please note that this only applies to actual time fields in the database file: numeric and alphanumeric fields which contain times cannot be identified as times by ***Slipstream***.

***DBF**	(Default). The format of the time is determined by the time format specified for the time field in the database file (DDS TIMFMT and TIMSEP keywords).
***EXCEL**	The time is converted to Excel format, i.e. a numeric value representing a number of seconds. This format is ideal if you are going to load the CSV file into Excel or a similar application. Excel will recognize the data as a time and allow time functions and editing to be applied to it.
***ISO**	ISO format (HH.MM.SS) is used.
***EUR**	European format (HH.MM.SS) is used.
***USA**	USA format (HH:MM:SS) is used.
***JIS**	Japanese Industrial Standard (HH:MM:SS) is used.
***HMS**	HHMMSS format.

Time Separator

***SYSVAL**	The time separator defined by system value QTIMFMT is used.
***JOB**	The time separator character defined in the job attributes is used.
***NONE**	No time separator character is used.
***PERIOD**	A period (full stop) . is used.
***COMMA**	A comma , is used.
***COLON**	A colon : is used.
***BLANK**	A blank is used.
separator_char	Type the separator character to be used to separate the day, month and year portions of the date.

Decimal point character

This element allows you to specify the characters to be used to denote a decimal point when representing fields with one or more decimal places.

Options are:

***SYSVAL**	The decimal format defined by system value QDECFMT is used.
***JOB**	The decimal format defined in the job attributes is used.
***PERIOD**	A period (full stop). is used.
***COMMA**	A comma , is used.

Apply edit codes and words

Whether Slipstream edits numeric values with their associated edit code or edit word before outputting the data.

For example, is a packed decimal field contains the value -123456.78 and has an associated edit code of $M, and *YES is specified for this element, the field value will be output as **$123,456.78-** rather than 123456.78.

Options are:

***NO**	Edit codes and edit words are ignored
***YES**	Numeric fields with associated edit codes or edit words (including editing defined in a Query/400 query or QM/Query form) will be edited before being output.

Trim blank from char fields

Whether Slipstream trims leading and/or trailing blanks when outputting character values.

Options are:

***BOTH**	Leading and trailing blanks are trimmed from character fields before they are output.
***LEADING**	Only leading blanks are trimmed
***TRAILING**	Only trailing blanks are trimmed
***NONE**	No blanks are trimmed.

FIXED – Fixed text options

Parameter	**FIXED (CVTDBFSTMF)** **TEXT (CVTDBFTXT)**
Description	**Specifies options for fixed-width ASCII text output**
Applies to commands:	**CVTDBFTXT** **CVTDBFSTMF (deprecated)**
Dependent on:	**TOFMT(*FIXED) (CVTDBFSTMF only)**
Supports Slipstream variables	**No**
Migration notes	**The parameter of CVTDBFTXT corresponding to the FIXED parameter of the CVTDBFSTMF command is called TEXT for consistency with the name of the command and the TEXT parameter of CoolSpools's CVTSPLTXT command etc. Its format has also been modified somewhat.** **Specifically, the CVTDBFSTMF FIXED parameter allows the use of a single value *DFT which is not available on the CVTDBFTXT TEXT parameter. Also, on the FIXED parameter of the CVTDBFSTMF command, the date format and date separator formed a related pair for which a single value could be specified, as did the time format and time separator. This tended to make use of this parameter excessively complicated. The TEXT parameter of the CVTDBFTXT command has been re-designed to improve ease of use, but code changes may be necessary when migrating from CVTDBFSTMF ...TOFMT(*FIXED) to use of the CVTDBFTXT command.**

Specifies options related to fixed-width text output.

Record delimiter

This element allows you to specify the characters to be used to indicate the end of a record in the text file.

Options are:

***CRLF**	Carriage return and line feed. Both a carriage return (x'0D') and a line feed (x'0A') character are used.
***CR**	Just a carriage return (x'0D') is used.
***LF**	Just a line feed (x'0A') is used.

Date format

This option controls the format in which date information in the database file is presented in the stream file. Please note that this only applies to actual date fields in the database file:

numeric and alphanumeric fields which contain date information cannot be identified as dates by ***Slipstream***.

***DBF**	(Default). The format of the date is determined by the date format specified for the date field in the database file (DDS DATFMT and DATSEP keywords).
***EXCEL**	The date is converted to Excel format, i.e. a numeric value representing a day count. This format is ideal if you are going to load the CSV file into Excel or a similar application. Excel will recognize the data as a date and allow date functions and editing to be applied to it.
***ISO**	ISO format (YYYY-MM-DD) is used.
***EUR**	European format (DD.MM.YYYY) is used.
***USA**	USA format (MM/DD/YYYY) is used.
***JIS**	Japanese Industrial Standard (YYYY-MM-DD) is used.
***SYSVAL**	The date format defined in system value QDATFMT is used.
***JOB**	The date format defined in the job attributes is used.
***DMY**	DDMMYY format.
***DMYY**	DDMMYYYY format.
***MDYY**	MMDDYYYY format.
***YYMD**	YYYYMMDD format.
***CDMY**	CDDMMYY format. The C indicates the century (0 = 20th, 1=21st)
***CMDY**	CMMDDYY format. The C indicates the century (0 = 20th, 1=21st)
***CYMD**	CYYMMDD format. The C indicates the century (0 = 20th, 1=21st)
***JUL**	YYDDD format
***LONGJUL**	YYYYDDD format

Date Separator

The character used to separate the different part of a date value.

***JOB**	The date separator character defined in the job attributes is used.
***NONE**	No date separator character is used.
***SLASH**	A forward slash (oblique or solidus) / is used.
***HYPHEN**	A hyphen (dash) – is used
***PERIOD**	A period (full stop) . is used.
***COMMA**	A comma , is used.
***COLON**	A colon : is used.
***BLANK**	A blank is used.
separator_char	Type the separator character to be used to separate the day, month and year portions of the date.

Time format

This option controls the format in which time information in the database file is presented in the stream file. Please note that this only applies to actual time fields in the database file: numeric and alphanumeric fields which contain times cannot be identified as times by ***Slipstream***.

You can specify one of the following single values for this option:

***DBF**	(Default). The format of the time is determined by the time format specified for the time field in the database file (DDS TIMFMT and TIMSEP keywords).
***EXCEL**	The time is converted to Excel format, i.e. a numeric value representing a number of seconds. This format is ideal if you are going to load the CSV file into Excel or a similar application. Excel will recognize the data as a time and allow time functions and editing to be applied to it.
***ISO**	ISO format (HH.MM.SS) is used.
***EUR**	European format (HH.MM.SS) is used.
***USA**	USA format (HH:MM:SS) is used.
***JIS**	Japanese Industrial Standard (HH:MM:SS) is used.
***HMS**	HHMMSS format.

Time Separator

The character used to separate the different part of a time value.

Options are:

***SYSVAL**	The time separator defined by system value QTIMFMT is used.
***JOB**	The time separator character defined in the job attributes is used.
***NONE**	No time separator character is used.
***PERIOD**	A period (full stop) . is used.
***COMMA**	A comma , is used.
***COLON**	A colon : is used.
***BLANK**	A blank is used.
separator_char	Type the separator character to be used to separate the day, month and year portions of the date.

Decimal point character

This element allows you to specify the characters to be used to denote a decimal point when representing fields with one or more decimal places.

Options are:

***SYSVAL**	The decimal format defined by system value QDECFMT is used.
***JOB**	The decimal format defined in the job attributes is used.
***PERIOD**	A period (full stop). is used.
***COMMA**	A comma , is used.

Field delimiter

The character value (if any) output between fields. Typically, when outputting a fixed-width ASCII text file, no field delimiter is required as each field always occupies the same positions in the record, but you can output a separator value to make the file easier to read if you wish.

Options are:

***NONE**	No delimiter is output between fields.

***SPACE**	A space character (ASCII x'20')
***TAB**	A tab character (ASCII x'09')
***PIPE**	A pipe character (| ASCII x'7C')
***SEMICOLON**	A semicolon (; ASCII x'3B')
***COMMA**	A comma (, ASCII x'2C')
sep_character	Specify the character to use as the separator

Suppress leading zeros

Whether Slipstream replaces leading zeros with blanks before outputting the data.

Numeric fields are always output at their full possible width, so that the values for a field in different records will line up on their decimal points, with a sign character at the beginning.

Options are:

***NO**	Leading zeros are retained
***YES**	Leading zeros are replaced by blanks.

Apply edit codes and words

Whether Slipstream edits numeric values with their associated edit code or edit word before outputting the data.

For example, is a packed decimal field contains the value -123456.78 and has an associated edit code of $M, and *YES is specified for this element, the field value will be output as **$123,456.78-** rather than 123456.78.

Options are:

***NO**	Edit codes and edit words are ignored
***YES**	Numeric fields with associated edit codes or edit words (including editing defined in a Query/400 query or QM/Query form) will be edited before being output.

Positive sign

The character value used in the sign position for positive fields.

Numeric fields are always output at their full possible width, so that the values for a field in different records will line up on their decimal points, with a sign character at the beginning.

When the value is negative, a minus sign (ASCII x'2D') will appear in the sign character position. This option determines the character that appears in that position for a positive value.

Options are:

***SPACE**	A space character (ASCII x'20')
***PLUS**	A plus sign (+ ASCII x'2B')
pos_sign	Specify the character to use as the positive sign

HTML – HTML options

Parameter	**HTML**
Description	**Specifies options for HTML output**
Applies to commands:	**CVTDBFHTML** **CVTDBFSTMF (deprecated)**
Dependent on:	**TOFMT(*HTML) (CVTDBFSTMF only)**
Supports Slipstream variables	**Yes (free-format text elements)**
Migration notes	**The format of the HTML parameter of the CVTDBFHTML command has been modified slightly from that of the CVTDBFSTMF command. Specifically, the CVTDBFSTMF HTML parameter allows the use of a single value *DFT which is not available on the CVTDBFHTML HTML parameter. Also, on the HTML parameter of the CVTDBFSTMF command, the date format and date separator formed a related pair for which a single value could be specified, as did the time format and time separator. This tended to make use of this parameter excessively complicated. The HTML parameter of the CVTDBFHTML command has been re-designed to improve ease of use, but code changes may be necessary when migrating from CVTDBFSTMF ...TOFMT(*HTML) to use of the CVTDBFHTML command.**

Specifies options related to HTML output.

Date format

This option controls the format in which date information in the database file is presented in the HTML file. Please note that this only applies to actual date fields in the database file: numeric and alphanumeric fields which contain date information cannot be identified as dates by ***Slipstream***.

Options are:

***DBF**	(Default). The format of the date is determined by the date format specified for the date field in the database file (DDS DATFMT and DATSEP keywords).
***EXCEL**	The date is converted to Excel format, i.e. a numeric value representing a day count. This format is ideal if you are going to load the CSV file into Excel or a similar application. Excel will recognize the data as a date and allow date functions and editing to be applied to it.
***ISO**	ISO format (YYYY-MM-DD) is used.
***EUR**	European format (DD.MM.YYYY) is used.
***USA**	USA format (MM/DD/YYYY) is used.
***JIS**	Japanese Industrial Standard (YYYY-MM-DD) is used.

***SYSVAL**	The date format defined in system value QDATFMT is used.
***JOB**	The date format defined in the job attributes is used.
***DMY**	DDMMYY format.
***DMYY**	DDMMYYYY format.
***MDYY**	MMDDYYYY format.
***YYMD**	YYYYMMDD format.
***CDMY**	CDDMMYY format. The C indicates the century (0 = 20th, 1=21st)
***CMDY**	CMMDDYY format. The C indicates the century (0 = 20th, 1=21st)
***CYMD**	CYYMMDD format. The C indicates the century (0 = 20th, 1=21st)
***JUL**	YYDDD format
***LONGJUL**	YYYYDDD format

Date separator

***JOB**	The date separator character defined in the job attributes is used.
***NONE**	No date separator character is used.
***SLASH**	A forward slash (oblique or solidus) / is used.
***HYPHEN**	A hyphen (dash) – is used
***PERIOD**	A period (full stop) . is used.
***COMMA**	A comma , is used.
***COLON**	A colon : is used.
***BLANK**	A blank is used.
separator_char	Type the separator character to be used to separate the day, month and year portions of the date.

Time format

This option controls the format in which time information in the database file is presented in the HTML file. Please note that this only applies to actual time fields in the database file: numeric and alphanumeric fields which contain times cannot be identified as times by ***Slipstream***.

You can specify one of the following single values for this option:

***DBF**	(Default). The format of the time is determined by the time format specified for the time field in the database file (DDS TIMFMT and TIMSEP keywords).
***EXCEL**	The time is converted to Excel format, i.e. a numeric value representing a number of seconds. This format is ideal if you are going to load the CSV file into Excel or a similar application. Excel will recognize the data as a time and allow time functions and editing to be applied to it.
***ISO**	ISO format (HH.MM.SS) is used.
***EUR**	European format (HH.MM.SS) is used.
***USA**	USA format (HH:MM:SS) is used.
***JIS**	Japanese Industrial Standard (HH:MM:SS) is used.
***HMS**	HHMMSS format.

Time separator

***SYSVAL**	The time separator defined by system value QTIMFMT is used.
***JOB**	The time separator character defined in the job attributes is used.
***NONE**	No time separator character is used.
***PERIOD**	A period (full stop) . is used.
***COMMA**	A comma , is used.
***COLON**	A colon : is used.
***BLANK**	A blank is used.
separator_char	Type the separator character to be used to separate the day, month and year portions of the date.

Decimal point character

This element allows you to specify the characters to be used to denote a decimal point when representing fields with one or more decimal places.

Options are:

***SYSVAL**	The decimal format defined by system value QDECFMT is used.
***JOB**	The decimal format defined in the job attributes is used.
***PERIOD**	A period (full stop). is used.
***COMMA**	A comma , is used.

Table cell spacing

This attribute specifies how much space (measured in pixels) the browser should leave between the left side of the table and the left-hand side of the leftmost column, the top of the table and the top side of the topmost row, and so on for the right and bottom of the table. The attribute also specifies the amount of space to leave between cells.

Apply edit codes and words

Whether Slipstream edits numeric values with their associated edit code or edit word before outputting the data.

For example, is a packed decimal field contains the value -123456.78 and has an associated edit code of $M, and *YES is specified for this element, the field value will be output as **$123,456.78-** rather than 123456.78.

Options are:

***NO**	Edit codes and edit words are ignored
***YES**	Numeric fields with associated edit codes or edit words (including editing defined in a Query/400 query or QM/Query form) will be edited before being output.

Convert null fields to

How Slipstream outputs null fields.

Options are:

***NBSP**	The HTML table cell corresponding to a null field value will contain a single non-break space
***EMPTY**	The HTML table cell corresponding to a null field value will be empty. This can look strange with some browsers.

HTML title

Text input here will appear in the title bar of your browser when the HTML file is displayed.

***NONE**	(Default) The file will have no title.
Title_text	Up to 128 characters of title text.

Table caption

Text input here will appear as the HTML table caption. The HTML table caption is a piece of explanatory text displayed above a data table.

***NONE**	(Default) The table will have no caption.
Caption_text	Up to 128 characters of caption text.

XML – XML options

Parameter	**XML**
Description	**Specifies options for XML output**
Applies to commands:	**CVTDBFXML**
Dependent on:	
Supports Slipstream variables	**Yes (free-format text elements)**

Specifies options related to XML output.

Root element

Determines the name given to the root element of the XML document that Slipstream will generate.

Options are:

***FROMFILE**	(Default). The name is generated from the name of te file being converted.
***QUALFILE**	The name is generated from the qualified name (file name and library name) of the file being converted.
root_element	Specify the name you want Slipstream to use for the root element.

Row element name

Determines the name given to the row elements of the XML document that Slipstream will generate. One row element is generated for each record selected from the input file.

Options are:

***RCDMT**	(Default). The name is generated from the name of the record format being converted.
***QUALRCDFMT**	The name is generated from the qualified record format name (file name and record format name) of the record format being converted.
row_element	Specify the name you want Slipstream to use for the row element.

Cols as elements or attributes

Controls the way in which the fields of a record are converted.

The value specified here controls the default action for the file. You can override that action by using the FLDSTYLES parameter to specify a different action for a particular field or fields.

Options are:

***ELEMENT**	(Default). Each field will be converted as a sub-element of the row element.

***ATTRIBUTE**	Each field will be converted to an attribute of the row element.

Generate elem/attr names from

Specifies how Slipstream should name elements and attributes generated from fields in the input file.

The value specified here controls the default naming for the file. You can override this naming by using the FLDSTYLES parameter to specify the naming for a particular field or fields.

Options are:

***FLDNAM**	(Default). The element or attribute generated for each file will be named based on the field name.
***QUALFLD**	The element or attribute generated for each file will be named based on the qualified field name (field name, record format name, file name).
***ALIAS**	The element or attribute generated for each file will be named based on the field alias name.
***QUALALIAS**	The element or attribute generated for each file will be named based on the qualified alias name (alias name, record format name, file name).

End of line separator

The character to use as an end-of-line marker.

Options are:

***NONE**	(Default). No end-of-line marker will be used.
***CRLF**	Lines will end with a carriage return-linefeed pair.
***LF**	Lines will end with just a linefeed.
***CR**	Lines will end with just a carriage return.

Apply edit codes and words

Whether numeric fields that have editing associated with them (edit code or edit words in DSS, or editing specified in Query/400 or a QM form), are edited before being output.

Options are:

***NO**	(Default). The editing is ignored and the field is written in its “raw” numeric form.
***YES**	Editing is applied and the field is written in edited form.

Null fields

How null fields are handled.

Options are:

***OMIT**	(Default). Null fields are not output. Where a field is null for a particular row, there will be no sub-element or attribute corresponding to that field in the row element.
***NIL**	Where a field is being converted as a sub-element, and that field is null for a particular row, the sub-element will be output with the attribute **xsi:nil="true"**. Where a field is being output as an attribute of the row element, the

attribute will be omitted from the row element (same as *OMIT).

Convert blank char fields as

How character fields that are all blanks handled.

Options are:

***EMPTY**	Blank fields are output as an empty element or attribute.
***KEEP**	Blank fields are output as an element or attribute whose value is one or more space characters.
***OMIT**	Blank fields are not output. Where a field contains all blanks for a particular row, there will be no sub-element or attribute corresponding to that field in the row element.
***NIL**	Where a field is being converted as a sub-element, and that field contains all blanks for a particular row, the sub-element will be output with the attribute **xsi:nil="true"**. Where a field is being output as an attribute of the row element, the attribute will be omitted from the row element (same as *OMIT).

Trim blanks from char fields

Whether blanks are trimmed from character fields before the value is output.

Options are:

***BOTH**	(Default). Leading and trailing blanks are trimmed.
***LEADING**	Just leading blanks are trimmed.
***TRAILING**	Just trailing blanks are trimmed.
***NONE**	No blanks are trimmed.

XMLNAMESPC – XML namespace options

Parameter	**XMLNAMESPC**
Description	**Specifies namespace-related options for XML output**
Applies to commands:	**CVTDBFXML**
Dependent on:	
Supports Slipstream variables	**No**

Specifies options related to namespaces for XML output.

Namespace URI

If a namespace URI is specified here, Slipstream will define that as the namespace for the document being created.

Options are:

***NONE**	(Default). No namespace will be specified.
namespace_URI	Specify the namespace URI to declare

Namespace prefix

Specifies the prefix which Slipstream should apply to all names in the document.

Options are:

***NONE**	(Default). No prefix is applied.
namespace_prefix	Specify the prefix you want Slipstream to apply to all names.

XMLSCHEMA – XML schema options

Parameter	**XMLSCHEMA**
Description	**Specifies schema-related options for XML output**
Applies to commands:	**CVTDBFXML**
Dependent on:	
Supports Slipstream variables	**No**

Specifies options related to styling for XML output.

There is a one single value:

***NONE**	(Default) No schema will be associated with the document.

Alternatively, specify options as listed below.

Schema type

Specifies how styling should be applied to the XML document.

Options are:

***XSD**	(Default). Slipstream will associate a XSD (XML Schemas Definition) schema with the document.
***DTD**	Slipstream will associate a DTD (Document Type Definitions) schema with the document.

Generate schema

Whether Slipstream should generate a simple XSD or DTD schema itself, or whether you will specify the name of an existing schema to use.

Options are:

***YES**	(Default). Slipstream will generate a simple schema itself.
***NO**	Slipstream will use an existing schema.

Replace existing file

When Slipstream is to generate a simple XSD or DTD schema itself, whether to replace any schema file that already exists.

Options are:

***NO**	(Default). If the schema file already exists, it will not be replaced and an error will occur.
***YES**	Slipstream will replace any existing schema file.

Schema

The path name of the existing schema file that will be used, or the path name of the schema file that Slipstream will generate.

Options are:

***TOSTMF**	(Default). The path name is the same as that specified on or derived from the TOSTMF parameter, with the extension changed appropriately (.XSD or .DTD).
schema_path	Specify the path name of the existing schema file to use or the schema file to be generated.

XMLSTYLING – XML styling options

Parameter	**XMLSTYLING**
Description	**Specifies styling-related options for XML output**
Applies to commands:	**CVTDBFXML**
Dependent on:	
Supports Slipstream variables	**No**

Specifies options related to styling for XML output.

There is a one single value:

***NONE** No styling will be applied to the document.

Alternatively, specify options as listed below.

Styling method

Specifies how styling should be applied to the XML document.

Options are:

***XSLT** (Default). Slipstream will use XSLT (Extensible Stylesheet Language Transformations) to apply styling to the XML document.

***CSS** Slipstream will use CSS (Cascading Stylesheets) to apply styling to the XML document.

Generate stylesheet

Whether Slipstream should generate a simple XSLT or CSS stylesheet itself, or whether you will specify the name of an existing stylesheet document to use.

Options are:

***YES** (Default). Slipstream will generate a simple stylesheet itself.

***NO** Slipstream will use an existing stylesheet

Replace existing file

When Slipstream is to generate a simple XSLT or CSS stylesheet itself, whether to replace any stylesheet that already exists.

Options are:

***NO** (Default). If the stylesheet already exists, it will not be replaced and an error will occur.

***YES** Slipstream will replace any existing stylsheet.

Stylesheet

The path name of the existing stylesheet that will be used, or the path name of the stylesheet that Slipstream will generate.

Options are:

***TOSTMF**	(Default). The path name is the same as that specified on or derived from the TOSTMF parameter, with the extension changed appropriately (.XSLT or .CSS).
stylesheet_path	Specify the path name of the existing stylesheet to use or the stylesheet to be generated.

CSSSTYLING – CSS stylesheet options

Parameter	**CSSSTYLING**
Description	**Specifies styling-related options for CSS styling**
Applies to commands:	**CVTDBFXML**
Dependent on:	**XMLSTYLING(*CSS)**
Supports Slipstream variables	**No**

Specifies options related to CSS styling for XML output.

There is a one single value:

***NONE**	No CSS styling options are defined.

Alternatively, specify options as listed below.

Generate :before selector

Specifies if and how a :before selector is generated.

Note that not all browsers currently support :before selectors (notably MS Internet Explorer does not).

Options are:

***AVAIL**	(Default). The :before selector is generated from the best available source, either the field text, alias name, column headings or field name.
***TEXT**	The :before selector is generated from the field text attribute (DDS TEXT).
***ALIAS**	The :before selector is generated from the field alias name (DDS ALIAS).
***COLHDG**	The :before selector is generated from the field column headings (DDS COLHDG).
***FLDNAM**	The :before selector is generated from the field name.

Padding width

Specifies the number of characters to which the :before text is padded. Note that this will only cause columns to line up if a monospace font (e.g. Courier New) is selected.

Options are:

50	(Default). 50 characters.
padding_width	Specify the width to which the :before text is padded.

Padding type

Specifies the how the :before text is padded.

Options are:

***DOTS**	(Default). The :before text is padded with dot leaders (. . .)

***SPACES**	The :before text is padded with spaces.
***NONE**	:before text is not padded to a fixed width

:before selector style

Free-form unvalidated text specifying CSS styling options to be applied to :before text.

Options are:

***NONE**	(Default). No styling options are applied.
CSS_styling	Specify a valid CSS style declaration for the :before text.

HEADER – Header row

Parameter	**HEADER**
Description	**Specifies options for generating a header row**
Applies to commands:	**CVTDBFXLS, CVTDBFCSV, CVTDBFHTML** **CVTDBFSTMF (deprecated)**
Dependent on:	**TOFMT(*XLS|*CSV|*HTML) (CVTDBFSTMF only)**
Supports Slipstream variables	**Yes (free-format text elements)**
Migration notes	**CVTDBFXLS includes some options on this parameter which are not available on CVTDBFCSV or CVTDBFHTML.**

Allows you to specify whether a head row should be created in the stream file and, if so, how.

Please note that a header row cannot be created for fixed-length ASCII files created with CVTDBFTXT or CVTDBFSTMF TOFMT(*FIXED) as the fixed-length nature of the columns in these types of file make the creation of header text impossible.

However, for CSV, Excel and HTML files it is common practice to include a single row at the beginning of the data which provides column labels for the data.

Source of header information

What attributes of the input file are used to generate the headings.

Options are:

AVAIL** (Default). ***Slipstream will select the best available labels from the input file to create the header row. This selection of label text is performed according to the following criteria. If the fields in the input file have Column Headings (DDS COLHDG keyword), they will be used to generate the column headings. Otherwise, if the fields in the input file have field aliases (DDS ALIAS keyword), those will be used instead. Otherwise, if the fields in the input file have text descriptors (DDS TEXT keyword), those will be used. Otherwise the field names will be used.

For fixed-length ASCII text files, no header row will be created.

***NONE** No header row is created.

***COLHDG** For CSV, Excel and HTML files, a header row is created from the field column headings (DDS COLHDG keyword). This option is invalid if TOFMT(*FIXED) is specified.

***ALIAS** For CSV, Excel and HTML files, a header row is created from the field aliases (DDS ALIAS keyword).

	This option is invalid if TOFMT(*FIXED) is specified.
***TEXT**	For CSV, Excel and HTML files, a header row is created from the field text descriptors (DDS TEXT keyword). This option is invalid if TOFMT(*FIXED) is specified.
***FLDNAM**	For CSV, Excel and HTML files, a header row is created from the field names. This option is invalid if TOFMT(*FIXED) is specified.
***COLHDG1**	The first column heading element only.
***COLHDG2**	The second column heading element only.
***COLHDG3**	The thrid column heading element only.
***COLHDG12**	Column heading elements 1 and 2 only.
***COLHDG13**	Column heading elements 1 and 3 only.
***COLHDG23**	Column heading elements 2 and 3 only.

Freeze Excel header line

Available on CVTDBFXLS and CVTDBFSTMF only.

Whether the header line is frozen in Excel (i.e. whether it scrolls with the rest of the data or remains fixed at the top of the screen while the data rows scroll).

Options are:

***YES**	(Default). The header row will be frozen and will not scroll with the data rows.
***NO**	The header row will be not frozen and will scroll with the data rows.

Line breaks in Excel headers?

Available on CVTDBFXLS and CVTDBFSTMF only.

Whether, when the header line is being constructed from DDS column heading values, line breaks are inserted between the different parts of the column headings so that the column headings wrap to multiple lines.

Options are:

***YES**	(Default). Line breaks are inserted between the different parts of the DDS column headings so that they appear one per line at the top of the Excel worksheet.
***NO**	A space is inserted between the different parts of the DDS column headings and Excel will control the wrapping of header text.

Additional header line 1

Additional header line 2

Additional header line 3

Three lines of free-format text which will appear at the top of the page above the column headings.

Headings on overflow sheets

Available on CVTDBFXLS and CVTDBFSTMF only.

Whether column headings are repeated at the top of any overflow worksheets that are necessary because the number of rows of data exceeds the maximum number of rows in an Excel worksheet.

Options are:

***YES**	(Default). Column headings appear at the top of each worksheet that is created.
***NO**	Column headings appear only at the top of the first worksheet that is created.

DFNSTYLES – Define styles

Parameter	**DFNSTYLES**
Description	**Defines styles which control the appearance of data on screen**
Applies to commands:	**CVTDBFXLS, CVTDBFHTML, CVTDBFXML**
Dependent on:	**None**
Supports Slipstream variables	**No**

Allows you to define the attributes of the default styles for different types of information or to created user-defined named styles which can be associated with particular fields by means of the related FLDSTYLES parameter. Styles control the appearance of data on screen. For example, they govern items such as:

- font attributes
- color
- numeric formatting

The precise set of attributes that can be controlled varies depending on the format of the data being created as some attributes are not relevant to certain output formats.

Slipstream styles defined will translate into Excel user-defined styles if converting to Excel format and CSS styles if converting to HTML/XML.

Style name

Each style is identified by means of a style name, which must be a valid OS/400 name up to 20 characters in length.

You can define your own named styles by choosing a name that is helpful to you, but there are also 7 pre-defined style names which have special meanings:

***DATA** The default style name for data rows.

If you specify *DATA for the name of the style, the attributes you specify will become the default attributes for data rows.

***HEADER** The default style name for header rows.

If you specify *HEADER for the name of the style, the attributes you specify will become the default attributes for header rows (rows generated as a result of the HEADER parameter settings).

***TITLE** The default style name for title rows.

If you specify *TITLE for the name of the style, the attributes you specify will become the default attributes for title rows. Title rows are those generated from the

additional heading lines elements of the HEADER parameter and the caption text of the HTML parameter.

***SUBTOTAL** The default style name for subtotal rows.

If you specify *SUBTOTAL for the name of the style, the attributes you specify will become the default attributes for subtotal rows. Subtotal rows are those that result from subtotals and group-by fields in Query/400 queries when the *COMBINED output form is selected.

***TOTAL** The default style name for total rows.

If you specify *TOTAL for the name of the style, the attributes you specify will become the default attributes for total rows. Total rows are those that result from subtotals and group-by fields in Query/400 queries when the *COMBINED output form is selected.

***ROOT** **XML Only.** The default style for the root element.

***ROW** **XML Only.** The default style for the row element, i.e. the element corresponding to a record in the input file.

If these styles are not defined, the default attributes assigned are as shown in the table below:

Attribute	*DATA	*HEADER	*TITLE	*SUBTOTAL	*TOTAL	*ROOT	*ROW
Locked (Excel only)	Yes	Yes	Yes	Yes	Yes	N/A	N/A
Hidden (Excel only)	No	No	No	No	No	N/A	N/A
Horizontal alignment	General	General	General	General	General	General	General
Indent	0	0	0	0	0	N/A	N/A
Vertical alignment	Top	Bottom	Top	Top	Top	Top	Top
Wrap text	No	Yes	No	No	No	No	No
Shrink to fit (Excel only)	No	Yes	No	No	No	N/A	N/A
Vertical alignment	Top	Top	Top	Top	Top	Top	Top
Row height	*AUTOFIT	*AUTOFIT	*AUTOFIT	*AUTOFIT	*AUTOFIT	*AUTOFIT	*AUTOFIT
Font name (Excel)	Arial	Arial	Arial	Arial	Arial	N/A	N/A

Font name (HTML & XML)	sans-serif	sans-serif	sans-serif	sans-serif	sans-serif	sans-serif	sans-serif
Font size in point (Excel)	10	10	10	10	10	N/A	N/A
Font size in point (HTML & XML)	12	12	12	12	12	12	12
Bold	No	Yes	Yes	Yes	Yes	Yes	Yes
Italic	No	No	No	No	No	No	No
Underlined	No	No	No	No	No	No	No
Text color	Black	Black	Black	Black	Black	Black	Black
Background color	White	White	White	White	White	White	White
Pattern color (Excel only)	*AUTO	*AUTO	*AUTO	*AUTO	*AUTO	*AUTO	*AUTO
Pattern style (Excel only)	*NONE	*NONE	*NONE	*NONE	*NONE	*NONE	*NONE
Border style (Excel)	*NONE	*NONE	*NONE	*NONE	*NONE	N/A	N/A
Border style (HTML)	*INSET	*INSET	*INSET	*INSET	*INSET	N/A	N/A
Border width (HTML only)	1	1	1	1	1	1	1
Border color	*AUTO	*AUTO	*AUTO	*AUTO	*AUTO	*AUTO	*AUTO
Number format type (Excel only)	*DFT	*DFT	*DFT	*DFT	*DFT	N/A	N/A
Decimal places (Excel only)	*FIELD	*FIELD	*FIELD	*FIELD	*FIELD	N/A	N/A
Thousands separator (Excel only)	*FMT	*FMT	*FMT	*FMT	*FMT	N/A	N/A
Currency symbol (Excel only)	*FMT	*FMT	*FMT	*FMT	*FMT	N/A	N/A
Negative numbers (Excel only)	*FMT	*FMT	*FMT	*FMT	*FMT	N/A	N/A
Custom number format (Excel	*NONE	*NONE	*NONE	*NONE	*NONE	N/A	N/A

only)							
Cell padding (HTML only)	1	1	1	1	1	1	1
Additional style declaration (HTML only)	*NONE	*NONE	*NONE	*NONE	*NONE	*NONE	*NONE
Display option (XML only)	*BLOCK	*BLOCK	*BLOCK	*BLOCK	*BLOCK	*BLOCK	*BLOCK

Example:

CVTDBFXLS
...
DFNSTYLES((HIGHLIGHT *YES *NO *GENERAL *NONE *BOTTOM *NO *NO *AUTOFIT *ARIAL 12 *YES *NO *NO *YELLOW *BLUE *AUTO *NONE *THIN))
FLDSTYLES((BALANCE *FLDNAM HIGHLIGHT))

This code defines a new style called “highlight” that uses Arial bold 12-point yellow on blue and applies that style to the field called “BALANCE”.

Example:

CVTDBFXLS
...
DFNSTYLES((*DATA *YES *NO *GENERAL *NONE *BOTTOM *NO *NO *AUTOFIT *COURIER 12 *NO *NO *NO *RED *SILVER *AUTO *NONE *THIN) (*HEADER *YES *NO *GENERAL *NONE *BOTTOM *YES *NO *AUTOFIT *COURIER 14 *YES *NO *NO *RED *SILVER *AUTO *NONE *THIN))

This code redefines the predefined *DATA and *HEADER styles and so modifies the default attributes for data rows to use Courier 12-point red on silver and the default attributes for column headings to use Courier 14-point bold red on silver.

Locked

(Excel only)

Whether cells to which this style is applied are locked when worksheet protection is in effect.

Options are:

***YES**	(Default). When the worksheet is protected, the cell will be locked (protected).
***NO**	When the worksheet is protected, the cell will remain unlocked.

Example:

CVTDBFXLS
...
DFNSTYLES((*DATA *NO) (*HEADER *YES *NO *GENERAL *NO *BOTTOM *ARIAL 10 *NO *NO *NO *WHITE *BLUE) (LOCKED *YES *NO *GENERAL *NO * BOTTOM *ARIAL 10 *NO *NO *NO *WHITE *BLUE)) FLDSTYLES((CUSTNO *FLDNAM LOCKED)) XLSPROTECT(*YES)

This code redefines the *DATA style such that, by default, data cells are not locked when worksheet protection is in effect. It then redefines the *HEADER style so that headers are locked and appear white on blue. Finally, it defines a new style called “LOCKED” such that cells to which this style is applied are locked and also appear white on blue. The style LOCKED is applied to the single column “CUSTNO” and worksheet protection is switched on.

The overall effect is to create a worksheet where the user can changes the data apart from the headings and the customer number column, which appear white on blue rather than black on white to emphasize the fact they are different.

Hidden

(Excel only)

Allows you to indicate that a column should be hidden. This might be useful if you do not wish the column to appear but want it to be available for calculations.

Options are:

***NO**	(Default). The column is not hidden.
***YES**	The columns will be hidden

Horizontal alignment

Controls the horizontal alignment of data in a cell.

Options are:

***GENERAL**	(Default). Character data is left-aligned. Numeric data and dates are right-aligned. In relation to header text, the alignment is dictated by the nature of the data in the column, not the header text, i.e. headings for columns of character data will align to the left and headings for numeric columns and date columns will align to the right.
***LEFT**	Left-aligned.
***RIGHT**	Right-aligned.
***CENTER**	Center-aligned
***FILL**	(Excel only) Fill. Repeats the data in the cell across the entire width of the column.
***JUSTIFY**	Forces data to fill the entire width of the column, wrapping text to additional lines, if necessary.
***DISTRIBUTED**	(Excel only) Distributed. Available only in Excel 2002 and above. It results in the cell contents being distributed across the width of the cell, to line up with both the left and right side.

Indent

Sets the text indent level. The effects of this are somewhat different between Excel and HTML/XML.

Options are:

***NONE**	(Default). No indent is applied.
0-15	(Excel) Sets the indentation level. Each indentation level is equivalent to 3 spaces. All text affected is indented to the same extent, i.e. where text wraps to more than one line, it is all indented to the same point.
0-99	(HTML/XML). Sets the text-indent property in ems (the width of an em is equivalent to the point size of the font). The first line of the text only is indented.

Vertical alignment

Controls the vertical alignment of data in a cell.

Options are:

***BOTTOM**	(Default). Information is aligned at the top of the cell.
***TOP**	Information is aligned at the bottom of the cell.
***CENTER**	information is aligned in the center of the cell.
***JUSTIFY**	Text is spread evenly vertically across the height of the cell.
***DISTRIBUTED**	(Excel only) Text is spread evenly between the top of the cell and the bottom. Effectively, blank space is placed between each line so that the complete cell is filled.

Wrap text

Controls whether text wraps in cells.

Options are:

***NO**	(Default). Text does not wrap in the cell. If the text does not fit in the column width, it is truncated.
***YES**	Text wraps in the cell. If the text foes not fit in the column width, it will flow on to multiple lines.

Shrink to fit

(Excel only)

Determines whether the cell contents are shrunk to fit the available column width by reducing the font size.

Options are:

***NO**	(Default). Text is not shrunk to fit.
***YES**	Text is fitted to the available column width by reducing the font size, as required.

Row height

Sets the height of rows.

Note that this attribute is only effective is set on one of the predefined styles:

- *DATA (controlling the height of data rows)
- *HEADER (controlling the height of the column headings row)
- *TITLE (controlling the height of title rows)

- *SUBTOTAL (controlling the height of subtotal rows)
- *TOTAL (Controlling the height of total rows)

Even if you associated every column with a style other than these, the row height set for that row will not be effective as the row height is always set from the appropriate predefined style from the list above.

Options are:

***AUTOFIT**	(Default). The height of rows is automatically set by Excel or your browser (HTML/XML) based on the font size.
0-409	(Excel) Specify the row height in points (72 points = 1 inch)
0-32767	(HTML/XML) Specify the row height in points (72 points = 1 inch)

Font name

Specifies the name of the font to be used.

Note that Slipstream cannot validate whether the font name you have specified is valid or whether it will be available when the file is opened. If the font name is typed incorrectly or if the font is not available when the file is opened, Excel or your browser will substitute a different font.

Note also that when the font you use in Excel is not one of the “well known” fonts (Arial, Courier New or Times New Roman), Slipstream may not be able to calculate column widths correctly because it has no access to the font metrics on which those calculations depend.

Excel options are:

***ARIAL**	(Default). Arial
***COURIER**	Courier New
***TIMES**	Times New Roman
font_name	Specify the name of the font to use

HTML/XML options are:

***SANS**	(Default) Sans-serif font family.
***SERIF**	Serif font family
***MONO**	Monospaced font family.
***ARIAL**	(Default). Arial
***COURIER**	Courier New
***TIMES**	Times New Roman
font_name	Specify the name of the font to use

Font size in points

The point size of the font to use. The default is 10 for Excel and 12 for HTML/XML.

Bold

Whether the font is bold or not. Note that setting this attribute will only result in a bold font if a suitable bold version of the font is available or if the normal font can be adapted.

Options are:

***NO**	(Default). Normal font

***YES**	Bold font.

Italic

Whether the font is italic or not. Note that setting this attribute will only result in an italic font if a suitable italic version of the font is available or if the normal font can be adapted.

Options are:

***NO**	(Default). Normal font
***YES**	Italic font.

Underlined

Whether the font is underlined or not and, if it is, the style of underlining.

Excel options are:

***NO**	(Default). No underlining
***SINGLE**	Single underlining
***DOUBLE**	Double underlining
***SGLACC**	Single accounting underlining
***DBLACC**	Double accounting underlining

HTML/XML options are:

***NO**	(Default). No underlining
***YES**	Single underlining

Text color

Determines the color of text in cells.

The Excel default is:

***AUTO**	The Excel default text color (usually black)

Alternatively, you can use one of the 56 built-in Excel colors listed below with their RGB coding.

*BLACK	000000
*WHITE	FFFFFF
*RED	FF0000
*LIME	00FF00
*BLUE	0000FF
*YELLOW	FFFF00
*FUCHSIA	FF00FF
*AQUA	00FFFF
*MAROON	800000
*GREEN	008000
*NAVY	000080
*OLIVE	808000
*PURPLE	800080
*TEAL	008080
*SILVER	C0C0C0
*GRAY	808080
*MAUVE	9999FF

*PLUM	993366
*YELLOWWHITE	FFFFCC
*PALETEAL	CCFFFF
*DARKFUCHSIA	660066
*BLUSH	FF8080
*MEDIUMBLUE	0066CC
*PALEMAUVE	CCCCFF
*NAVY	000080
*FUCHSIA	FF00FF
*YELLOW	FFFF00
*AQUA	00FFFF
*PURPLE	800080
*MAROON	800000
*TEAL	008080
*BLUE	0000FF
*SKYBLUE	00CCFF
*PALETURQUOISE	CCFFFF
*PALEGREEN	CCFFCC
*PALEYELLOW	FFFF99
*DARKAQUA	99CCFF
*ROSE	FF99CC
*LAVENDER	CC99FF
*TAN	FFCC99
*LIGHTBLUE	3366FF
*TURQUOISE	33CCCC
*PALELIME	99CC00
*GOLD	FFCC00
*LIGHTORANGE	FF9900
*ORANGE	FF6600
*BLUEGRAY	666699
*PALEGRAY	969696
*DARKTEAL	003366
*SEAGREEN	339966
*DARKGREEN	003300
*DARKOLIVE	333300
*BROWN	993300
*PLUM	993366
*INDIGO	333399
*DARKGRAY	333333

When converting to *XLSX format, you can also optionally specify your own RGB color code in the form of three hexadecimal digits (similar to the codes shown in the table above). Please note that this option is not supported when converting to BIFF8 format.

The HTML/XML default is:

***BLACK**	Black

Alternatively, you can use one of the HTML colors listed below with their RGB coding.

*ALICEBLUE	F0F8FF
*ANTIQUEWHITE	FAEBD7

*AQUA	00FFFF
*AQUAMARINE	7FFFD4
*AZURE	F0FFFF
*BEIGE	F5F5DC
*BISQUE	FFE4C4
*BLACK	000000
*BLANCHEDALMOND	FFEBCD
*BLUE	0000FF
*BLUEVIOLET	8A2BE2
*BROWN	A52A2A
*BURLYWOOD	DEB887
*CADETBLUE	5F9EA0
*CHARTREUSE	7FFF00
*CHOCOLATE	D2691E
*CORAL	FF7F50
*CORNFLOWERBLUE	6495ED
*CORNSILK	FFF8DC
*CRIMSON	DC143C
*CYAN	00FFFF
*DARKBLUE	00008B
*DARKCYAN	008B8B
*DARKGOLDENROD	B8860B
*DARKGRAY	A9A9A9
*DARKGREY	A9A9A9
*DARKGREEN	006400
*DARKKHAKI	BDB76B
*DARKMAGENTA	8B008B
*DARKOLIVEGREEN	556B2F
*DARKORANGE	FF8C00
*DARKORCHID	9932CC
*DARKRED	8B0000
*DARKSALMON	E9967A
*DARKSEAGREEN	8FBC8F
*DARKSLATEBLUE	483D8B
*DARKSLATEGRAY	2F4F4F
*DARKSLATEGREY	2F4F4F
*DARKTURQUOISE	00CED1
*DARKVIOLET	9400D3
*DEEPPINK	FF1493
*DEEPSKYBLUE	00BFFF
*DIMGRAY	696969
*DIMGREY	696969
*DODGERBLUE	1E90FF
*FELDSPAR	D19275
*FIREBRICK	B22222
*FLORALWHITE	FFFAF0
*FORESTGREEN	228B22
*FUCHSIA	FF00FF
*GAINSBORO	DCDCDC
*GHOSTWHITE	F8F8FF
*GOLD	FFD700
*GOLDENROD	DAA520

*GRAY	808080
*GREY	808080
*GREEN	008000
*GREENYELLOW	ADFF2F
*HONEYDEW	F0FFF0
*HOTPINK	FF69B4
*INDIANRED	CD5C5C
*INDIGO	4B0082
*IVORY	FFFFF0
*KHAKI	F0E68C
*LAVENDER	E6E6FA
*LAVENDERBLUSH	FFF0F5
*LAWNGREEN	7CFC00
*LEMONCHIFFON	FFFACD
*LIGHTBLUE	ADD8E6
*LIGHTCORAL	F08080
*LIGHTCYAN	E0FFFF
*LIGHTGOLDENROD	FAFAD2
*LIGHTGRAY	D3D3D3
*LIGHTGREY	D3D3D3
*LIGHTGREEN	90EE90
*LIGHTPINK	FFB6C1
*LIGHTSALMON	FFA07A
*LIGHTSEAGREEN	20B2AA
*LIGHTSKYBLUE	87CEFA
*LIGHTSLATEBLUE	8470FF
*LIGHTSLATEGRAY	778899
*LIGHTSLATEGREY	778899
*LIGHTSTEELBLUE	B0C4DE
*LIGHTYELLOW	FFFFE0
*LIME	00FF00
*LIMEGREEN	32CD32
*LINEN	FAF0E6
*MAGENTA	FF00FF
*MAROON	800000
*MEDIUMAQUAMARINE	66CDAA
*MEDIUMBLUE	0000CD
*MEDIUMORCHID	BA55D3
*MEDIUMPURPLE	9370D8
*MEDIUMSEAGREEN	3CB371
*MEDIUMSLATEBLUE	7B68EE
*MEDIUMSPRINGGREEN	00FA9A
*MEDIUMTURQUOISE	48D1CC
*MEDIUMVIOLETRED	C71585
*MIDNIGHTBLUE	191970
*MINTCREAM	F5FFFA
*MISTYROSE	FFE4E1
*MOCCASIN	FFE4B5
*NAVAJOWHITE	FFDEAD
*NAVY	000080
*OLDLACE	FDF5E6
*OLIVE	808000

*OLIVEDRAB	6B8E23
*ORANGE	FFA500
*ORANGERED	FF4500
*ORCHID	DA70D6
*PALEBLUE	ADD8E6
*PALEBROWN	CD853F
*PALECYAN	E0FFFF
*PALEGOLDENROD	EEE8AA
*PALEGRAY	D3D3D3
*PALEGREY	D3D3D3
*PALEGREEN	98FB98
*PALEMAG	DDA0DD
*PALETURQUOISE	AFEEEE
*PALEVIOLETRED	D87093
*PALEYLW	FFFFE0
*PAPAYAWHIP	FFEFD5
*PEACHPUFF	FFDAB9
*PERU	CD853F
*PINK	FFC0CB
*PLUM	DDA0DD
*POWDERBLUE	B0E0E6
*PURPLE	800080
*RED	FF0000
*ROSYBROWN	BC8F8F
*ROYALBLUE	4169E1
*SADDLEBROWN	8B4513
*SALMON	FA8072
*SANDYBROWN	F4A460
*SEAGREEN	2E8B57
*SEASHELL	FFF5EE
*SIENNA	A0522D
*SILVER	C0C0C0
*SKYBLUE	87CEEB
*SLATEBLUE	6A5ACD
*SLATEGRAY	708090
*SLATEGREY	708090
*SNOW	FFFAFA
*SPRINGGREEN	00FF7F
*STEELBLUE	4682B4
*TAN	D2B48C
*TEAL	008080
*THISTLE	D8BFD8
*TOMATO	FF6347
*TURQUOISE	40E0D0
*VIOLET	EE82EE
*VIOLETRED	D02090
*WHEAT	F5DEB3
*WHITE	FFFFFF
*WHITESMOKE	F5F5F5
*YELLOW	FFFF00
*YELLOWGREEN	9ACD32

You can also optionally specify your own RGB color code in the form of three hexadecimal digits (similar to the codes shown in the table above).

Background color

Determines the color of the background of a cell.

The Excel default is:

***AUTO**	The Excel default background color (usually white)

Alternatively, you can use the same Excel options as listed for text color above.

The HTML/XML default is:

***WHITE**	White

Alternatively, you can use the HTML color options as listed for text color above.

Pattern color

(Excel only)

Determines the color of the any pattern applied to a cell.

The Excel default is:

***AUTO**	The Excel default pattern color (usually black)

Alternatively, you can use the same Excel options as listed for text color above.

Pattern style

(Excel only)

Determines the styler of the any pattern applied to a cell.

The default is:

***NONE**	No pattern

The available pattern options are the following names, which correspond to Excel's builtin patterns:

***SOLID**
***GRAY75**
***GRAY50**
***GRAY25**
***GRAY12.5**
***GRAY6.25**
***HRZSTRIPE**
***VRTSTRIPE**
***REVERSEDIAGSTRIPE**
***DIAGSTRIPE**
***DIAGCROSSHATCH**

***THICKDIAGCROSSHATCH**
***THINHRZSTRIPE**
***THINVRTSTRIPE**
***THINREVERSEDIAGSTRIPE**
***THINDIAGSTRIPE**
***THINHRZCROSSHATCH**
***THINDIAGCROSSHATCH**

Border style (Excel)

Determines the style of the border around a cell.

The Excel default is:

***NONE**	No border

Other Excel options are the following list of names corresponding to Excel's builtin border styles:

***THIN**
***MEDIUM**
***DASHED**
***DOTTED**
***THICK**
***DOUBLE**
***HAIR**

The HTML/XML options correspond to the CSS border style options:

***INSET**	(Default) CSS inset border style
***DASHED**	
***DOTTED**	
***DOUBLE**	
***GROOVE**	
***HIDDEN**	
***OUTSET**	
***RIDGE**	
***SOLID**	

Border width

(HTML only)

The width of the cell border in pixels.

Border color

The color of the border. Options are the same as for text color above.

Number format type

(Excel only)

Sets the category of number formatting applied to numbers in cells to which this style relates. The following options allow you to modify or override aspects of the default formatting determined by your choice for this parameter element.

Options are:

***DFT**	Slipstream will decide the formatting to applied based on the editing associated with the field in question, i.e. any edit code or edit word defined in the field's DDS or any editing defined in a Query/400 query or QM form for the field.
***GENERAL**	Ignore any editing associated with the field and format numeric data with general numbers in them.
***FIXED**	Ignore any editing associated with the field and format numeric data with a fixed number of decimal places.
***CURRENCY**	Ignore any editing associated with the field and format numeric data as a currency amount.
***ACCOUNTING**	Ignore any editing associated with the field and format numeric data as an accounting value. The Accounting category is the same as the Currency category, except it will align currency symbols and decimal points.
***DATE**	Ignore any formatting associated with the field and format it as a date. If the field does not contain a valid date, it will be formatted according to any editing associated with the field.
***TIME**	Ignore any formatting associated with the field and format it as a time or date/time. If the field does not contain a valid time or timestamp, it will be formatted according to any editing associated with the field.
***PERCENT**	Ignore any editing associated with the field, multiply the value by 100 and format numeric data as a percentage.
***SCIENTIFIC**	Ignore any editing associated with the field, and format numeric data in scientific notation.
***TEXT**	Ignore any editing associated with the field, and format numeric data as text.
***CUSTOM**	Apply a custom number format specified on the custom number format element below.

Decimal places

(Excel only)

Where a numeric format (other than *DFT) that can include decimal places was specified on the number format type parameter, this parameter element determines the number of decimal places displayed.

Options are:

***FIELD**	The number of decimal places defined for the field in its DDS.
dec_places	Specify the number of decimal places

Thousands separator

(Excel only)

Where a numeric format (other than *DFT) that can include thousands separators was specified on the number format type parameter, this parameter element determines whether thousands separators actually appear.

Options are:

***FMT**	Whether thousands separators appear depends on the number format type selected. Accounting and currency formatting will include thousands separators but other types will not.
***YES**	Include thousands separators in the number format irrespective of the fact that the number format type specified does not normally include them. For example you can format percentage values with thousands separators using this option.
***NO**	Do not **i**nclude thousands separators in the number format irrespective of the fact that the number format type specified does normally include them. For example, you can format currency values without thousands separators using this option.

Currency symbol

(Excel only)

Where a numeric format (other than *DFT) that can include a currency symbol was specified on the number format type parameter, this parameter element determines whether a currency symbol actually appears and what that symbol should be.

Options are:

***FMT**	Whether a currency symbol appears depends on the number format type selected. Accounting and currency formatting will include a currency symbol but other types will not. The currency symbol will be derived from the system value QCURSYM.
***SYSVAL**	Include a currency symbol in the number format irrespective of the fact that the number format type specified does not normally include one. The currency symbol will be derived from the system value QCURSYM.
***NO**	Do not **i**nclude a currency symbol in the number format irrespective of the fact that the number format type specified does normally include one. You can use this option to display a currency value with no currency symbol.
currency_symbol	Include a currency symbol in all numbers. The currency symbol will be the one specified here.

Negative numbers

(Excel only)

Overrides the way in which negative numbers are displayed.

Options are:

***FMT**	The format of negative numbers is determined by the option specified for the number format type.
***LEADING**	A leading minus sign is displayed.
***TRAILING**	A trailing minus sign is displayed.
***PARENTHESES**	Negative numbers appear in parentheses.
***RED**	Negative numbers appear in red.
***REDL**	Negative numbers appear in red with a leading minus sign.
***REDT**	Negative numbers appear in red with a trailing minus sign.
***REDP**	Negative numbers appear in red and in parentheses.

Custom number format

(Excel only)

Specify a custom number format. *CUSTOM must be specified for the number format ype element above.

Options are:

***NONE**	No custom number format is defined.
number_format	Specify the custom Excel number format to use.

Cell padding

(HTML only)

The padding to apply to the cell, in pixels.

Additional style declaration

(HTML only)

A free-format, unvalidated string of text which will be appended to the style declaration generated by the previous elements. This option enables you to specify additional CSS formatting not available from this parameter. However, you must ensure that the text you enter is a valid portion of a CSS style declaration.

For example, specifying '**font-variant: small-caps**' would cause the text to appear in small capitals.

Display option (XML only)

Sets the CSS display style.

Options are:

***BLOCK**	(Default). Takes up the full width available, with a new line before and after.
***INLINE**	Takes up only as much width as it needs, and does not force new lines

CNDFMTGRP – Conditional formatting groups

Parameter	**CNDFMTGRP**
Description	**Defines groups of conditional formatting rules and the range of cells to which they apply**
Applies to commands:	**CVTDBFXLS**
Dependent on:	**EXCEL(*XLSX).**
Supports Slipstream variables	**No**

Specifies conditional formatting rule groups. A conditional formatting rule group defines a group of related rules which will be applied, in a given priority sequence, to a range of cells in order to determine the appearance of those cells. For example, you might define a rule that tests the value of a customer account balance field and make rows where the balance is negative red and those where it is above a certain level green etc.

Please note that conditional formatting is currently only available when EXCEL(*XLSX...) is specified, i.e. when converting an Office Open XML .xlsx (Excel 2007) document. We hope to add support for some aspects of conditional formatting when converting to .xls (Excel 97) format at some point in the future.

See the CNDFMTRULE parameter below for examples of how to define conditional formatting.

Rule group number

Specifies an arbitrary, non-zero, positive integer which identifies the rule group. You can choose any number you like to identify the group, but it must be unique for all rule groups defined on the command.

The rule group number is used to match rules defined on the CNDFMTRULE parameter against rule groups defined on the CNDFMTGRP parameter. The CNDFMTGRP parameter defines group-level attributes such as the range of cells to which the rules should be applied, whereas the CNDFMTRULE parameter defines the individual rules in the group that will tested, one after another, in the priority sequence you specify, against those cells.

Rule group name

Specifies an optional, arbitrary name which identifies the rule group. You can choose any name you like to identify the group. The name has no function other than to help you document and remember the purpose of a given rule group.

Apply to rows

Which rows in the worksheet the rules should be applied to.

Options are:

***ALL** (Default). The rules are applied to all rows in the worksheet, including those not populated by data. If new data is entered after the last row of data, the rules will apply to those new rows too.

***USED**	The rules apply only to the rows populated with data.

Identify columns by

How the columns the rules should apply to are identified on the following element:

Options are:

***FLDNAM**	(Default). The following parameter element will contain one or more field names from the input file that identify the columns in the worksheet to which the rules apply. The rules will apply to the columns corresponding to those fields.
***COLUMN**	The following parameter element will contain one or more column letter (A-Z, AA-ZZ etc.) identifying the columns in the worksheet to which the rules apply.

Apply to fields/columns

Identifies the columns in the worksheet to which the rules apply.

If **Identify columns by** above is *FLDNAM, names entered on this parameter element are interpreted as field names from the input file. The rules will apply to the corresponding columns in the worksheet.

If **Identify columns by** above is *COLUMN, names entered on this parameter element are interpreted as column letters A-Z, AA-ZZ etc.) The rules will apply to the columns thus identified in the worksheet.

You may enter up to 50 field names or column letters. The default is the single value ***ALL** indicating that the rules apply to all columns in the worksheet.

CNDFMTRULE – Conditional formatting rules

Parameter	**CNDFMTRULE**
Description	**Defines individual conditional formatting rules**
Applies to commands:	**CVTDBFXLS**
Dependent on:	**EXCEL(*XLSX).**
Supports Slipstream variables	**No**

Specifies conditional formatting rules.

A conditional formatting rule group defines a group of related rules which will be applied, in a given priority sequence, to a range of cells in order to determine the appearance of those cells. For example, you might define a rule that tests the value of a customer account balance field and make rows where the balance is negative red and those where it is above a certain level green etc.

The CNDFMTGRP parameter defines conditional formatting rule groups and group-level attributes such and the range of cells to which the rules in the group will be applied.

The CNDFMTRULE parameter defines the individual rules within those groups which are tested in turn.

Please note that conditional formatting is currently only available when EXCEL(*XLSX...) is specified, i.e. when converting an Office Open XML .xlsx (Excel 2007) document. We hope to add support for some aspects of conditional formatting when converting to .xls (Excel 97) format at some point in the future.

Rule group number

Specifies an arbitrary, non-zero, positive integer which identifies the rule group. You can choose any number you like to identify the group, but it must be unique for all rule groups defined on the command.

The rule group numb you specify here must correspond to the rule group number of a rule group defined on the CNDFMTGRP parameter. The CNDFMTGRP parameter defines group-level attributes such as the range of cells to which the rules should be applied, whereas the CNDFMTRULE parameter defines the individual rules in the group that will tested, one after another, in the priority sequence you specify, against those cells.

Rule priority

The priority of this conditional formatting rule. Where the group contains several rules, this value is used to determine which rule takes precedence and there which style is applied. Lower numeric values are higher priority than higher numeric values, where 1 is the highest priority.

Field to test

Specifies the field in the input file which is tested in order to determine if the rule should evaluate to true or false.

Options are:

***CELLIS** (Default). The logic test is carried out on each individual cell within the range of cells to which this rule is applied, not to any particular column.

For example, if you were using conditional formatting rules to apply different colors, and you specify a *CELLIS rule, each separate cell in the range of cells to which the rules apply will be colored differently depending on the value of those individual cells.

***FORMULA** You will specify a formula on the **Parameter value(s)** element below. That formula will determine whether the rule evaluates to true or false and therefore what styling is applied.

field_name Specify a field name from the input file. Slipstream will generate a formula which carries out the required logic test against the value of this particular field. For each row, the value of this field in that row will determine the formatting of cells to which this rule applies.

For example, if your file contains customer account details, and you wish to color the rows based on the value of the customer's account balance, you might specify the BALANCE field here as that is the field which determines how the rows should be formatted.

Test to apply

Specifies the logic test which is carried out to determine if rule evaluates to true or false and therefore what styling to apply.

Options are:

***NONE** (Default) None. Only valid if *FORMULA is specified or the previous element, i.e. you will specify your own formula to apply on the **Parameter value(s)** element below.

The following tests compare the value of field identified by the previous parameter element, or each individual cell (if *CELLIS was specified) , against the parameter value or values specified on the **Parameter value(s)** element below.

***EQ** Equal.

A single value must be specified on **Parameter value(s)** below. The rule is true if the field or cell value is equal to this value.

***GT** Greater than.

A single value must be specified on **Parameter value(s)** below. The rule is true if the field or cell value is greater than this value.

***LT** Less than.

A single value must be specified on **Parameter value(s)** below. The rule is true if the field or cell value is less than this value.

***GE** Greater than or equal to.

A single value must be specified on **Parameter value(s)** below. The rule is true if the field or cell value is greater than or equal to this value.

***LE** Less than or equal to.

A single value must be specified on **Parameter value(s)** below. The rule is true if the field or cell value is less than or equal to this value.

***NE** Not equal to.

A single value must be specified on **Parameter value(s)** below. The rule is true if the field or cell value is not equal to this value.

***BETWEEN** Between.

Two values must be specified on **Parameter value(s)** below, separated by one or more spaces. The rule is true if the field or cell value is greater than or equal to the first value and less than or equal to the second value.

***NOTBETWEEN** Not between.

Two values must be specified on **Parameter value(s)** below, separated by one or more spaces. The rule is true if the field or cell value is less than the first value or greater than the second value.

***CT** Contains.

A single value must be specified on **Parameter value(s)** below and it is interpreted as a text string. The rule is true if the field or cell value contains the text string specified

***CONTAINS** Same as *CT.

***NC** Not contains.

A single value must be specified on **Parameter value(s)** below and it is interpreted as a text string. The rule is true if the field or cell value does not contain the text string specified

***NOTCONTAINS** Same as *NC.

***BEGINSWITH** Begins with.

A single value must be specified on **Parameter value(s)** below and it is interpreted as a text string. The rule is true if the field or cell value begins with the text string specified

***ENDSWITH** Ends with.

A single value must be specified on **Parameter value(s)** below and it is interpreted as a text string. The rule is true if the field or cell value ends with the text string specified

***BLANKS** Contains blanks.

The value of the **Parameter value(s)** below is irrelevant and is ignored. The rule is true if the field or cell value is blank (is empty or contains only spaces).

***NOTBLANKS** Does not contain blanks.

The value of the **Parameter value(s)** below is irrelevant and is ignored. The rule is true if the field or cell value is not blank (is not empty or does not contain only spaces).

***TIMEPERIOD** Time period.

The value of the **Parameter value(s)** below must be one of the special time-period values listed below (*LASTMONTH etc.). The rule is true if the field or cell value is number which Excel can interpret as a date and that date matches the time period specified.

The following tests are only supported if *CELLIS was specified for the **Field to test** element. Each value of cell in the range covered by the rule group is tested individually.

***TOPN** Top n values.

The value of the **Parameter value(s)** below must be a number indicating the value of n. The rule is true if the field or cell value in the top n values.

***BOTTOMN** Bottom n values.

The value of the **Parameter value(s)** below must be a number indicating the value of n. The rule is true if the field or cell value in the bottom n values.

***TOPNPC** Top n percent.

The value of the **Parameter value(s)** below must be a number indicating the value of n. The rule is true if the field or cell value in the top n percent of values.

***BOTTOMNPC** Bottom n percent.

The value of the **Parameter value(s)** below must be a number indicating the value of n. The rule is true if the field or cell value in the bottom n percent of values.

***DUPLICATE** Duplicate values

The value of the **Parameter value(s)** below is irrelevant and is ignored. The rule is true if the field or cell value is not unique in the range.

***UNIQUE** Duplicate values

The value of the **Parameter value(s)** below is irrelevant and is ignored. The rule is true if the field or cell value is unique in the range.

Parameter value(s)

The parameter value or values required for the test defined above.

The interpretation of the parameter element is dependent on the value of the **Test to apply** element:

Value of **Test to apply**	Interpretation of **Parameter value(s)**
*EQ, *LT, *LE, *GT, *GE, *NE	A single value representing a number or string, e.g. 1000 New York
*BETWEEN, *NOTBETWEEN	A pair of values representing numbers or strings. Separate each value by one or more spaces, e.g.: 1000 2000 A Z
*CT, *NC, *CONTAINS, *NOTCONTAINS, *BEGINSWITH, *ENDSWITH	A single value representing a text string, e.g. New York

*TIMEPERIOD	Must be one of the special time-period values specified below, e.g. *LASTMONTH.
*TOPN, *BOTTOMN	The ranking value, e.g. 10 = “Top 10”
*TOPNPC, *BOTTOMNPC	The percentage value, e.g. 10 = “Top 10%”

When **Test to apply** is *TIMEPERIOD, the value must be one of the following special time periods:

***THIS MONTH**	This month. The date falls in the current calendar month.
***LASTMONTH**	Last month. The date falls in the previous calendar month.
***NEXTMONTH**	Next month. The date falls in the following calendar month.
***THISWEEK**	This week. The date falls in the current week.
***LASTWEEK**	Last week. The date falls in the previous week.
***NEXTWEEK**	Next week. The date falls in the next week.
***LAST7DAYS**	Last 7 days. The date falls in the last seven days.
***TODAY**	Today. The date is the current date
***YESTERDAY**	Yesterday. The date is one day prior to the current date.
***TOMORROW**	Tomorrow. The date is one day after the current date.

When **Field to test** is *FORMULA, you must specify a formula of your own on this parameter element. If the result of the formula is true, the style associated with this rule will be applied. When specifying cell references in your formula, the row number should correspond to the data row in the worksheet, taking account of column headings and additional heading rows. Use a relative column reference to test each cell in the range separately or an absolute column reference to test the value of a specific column. Do NOT precede the formula by an equals sign = as you might do in a cell.

There is one other special value: ***AVG.** This allows you test against the average value for the selected range. This is only permitted where:

- **Field to test** is *CELLIS
- **Test to apply** is *EQ, *GT, *LT, *LE or *GE

Apply style name

The name of the style to apply if the rule evaluates to true.

The style name must match the name of a style specified on the DFNSTYLES parameter.

Note that Excel does not allow all of the attributes that can be defined on the DFNSTYLES parameter to be controlled by conditional formatting. For example, while you can change the text color or make the text bold or italic, you cannot change the font name or font size. If you attempt to modify these using conditional formatting, Excel will ignore that change.

Stop if true

Determines whether Excel stops evaluating rules in the group as soon as one has evaluated to true or whether it carries on and checks the next rule.

If this flag is '1',

.Apply to rows

Which rows in the worksheet the rules should be applied to.

Options are:

***YES**	(Default). No rules with lower priority may be applied over this rule, when this rule evaluates to true
***NO**	Other rules with a lower priority will also be evaluated and may override aspects of the formatting.

Examples

The following examples assume that the DFNSTYLES parameter (not shown here for the sake of clarity) has been used to define styles called RED, ORANGE and YELLOW (which might set the cell colors to have a red, orange or yellow background, for example).

Example 1:

```
CVTDBFXLS
FROMFILE(CUSTACCT)
EXCEL(*XLSX)
...
CNDFMTGRP(     (1 BALANCES *USED *FLDNAM *ALL)
               (2 DUESOON *ALL *FLDNAM (DUEDATE)))
CNDFMTRULE(    (1 1 BALANCE *LT 0 RED *YES)
               (1 2 BALANCE *BETWEEN '0 100' ORANGE)
               (2 1 DUEDATE *TIMEPERIOD *NEXTMONTH YELLOW))
```

Here the customer accounts file is being converted to an Excel spreadsheet in Open Office XML (.xlsx) format.

Two groups of conditional formatting rules are defined:

- Group 1 (named “BALANCES”) which is applied to all columns

  This has two rules:

  - If the value of the BALANCE field is zero, the entire row will have the RED style applied to it
  - If the value of the BALANCE field is between 0 and 100, the entire row will have the ORANGE style applied to it.

- Group 2 (named “DUESOON”) which is applied just to the DUEDATE field

  This has a single rule:

  - If the DUEDATE field contains a date in the following calendar month, the YELLOW style is applied to the DUEDATE cell in the row.

Example 2:

```
...
CNDFMTGRP(     (1 ABOVE_AVERAGE  *USED *FLDNAM BALANCE))
CNDFMTRULE(    (1 1 BALANCE *GT *AVG RED *YES))
```

A single conditional formatting group is defined:

- Group 1 (named “ANOVE_AVERAGE”) which is applied just to the column corresponding to the field called BALANCE.

  This consists of a single rule:

  - If the value of the BALANCE field in the current row is greater than the average value of the BALANCE field in all rows, the balance field will have the RED style applied to it

Example 3:

```
...
CNDFMTGRP(     (1 TOP_10_PERCENT  *USED *FLDNAM BALANCE))
CNDFMTRULE(    (1 1 *CELLIS *TOPNPC 10 RED *YES))
```

A single conditional formatting group is defined:

- Group 1 (named “TOP_10_PERCENT”) which is applied just to the column corresponding to the field called BALANCE.

  This consists of a single rule:

  - If the value of the BALANCE field in the current row is in the top 10% of values in the BALANCE field for all rows, the balance field will have the RED style applied to it

Example 4:

```
...
CNDFMTGRP(     (1 ALERT_FLAG  *USED *FLDNAM *ALL))
CNDFMTRULE(    (1 1 ALERT *NOTBLANKS *NONE RED *YES))
```

A single conditional formatting group is defined:

- Group 1 (named “ALERT_FLAG”) which is applied to all columns.

  This consists of a single rule:

  - If the value of the ALERT field in the current row is not blanks, the entire row have the RED style applied to it

Example 5:

```
...
CNDFMTGRP(     (1 NORTH_AMERICA *USED *FLDNAM *ALL))
CNDFMTRULE((1 1 *FORMULA *NONE 'OR($J2="US",$J2="CA")' RED *YES))
```

A single conditional formatting group is defined:

- Group 1 (named “NORTH_AMERICA”) which is applied to all columns.

  This consists of a single rule:

  - A user-defined formula is applied. The formula checks the value of column J (assumed to contain a country code) and applies the RED style if the value is US or CA. Note that an absolute column reference ($J) is used and that the row reference is 2 (assuming a single row of column headings at the start of the sheet).

FLDSTYLES – Field styles and attributes

Parameter	**FLDSTYLES**
Description	**Let you override the style of individual fields and set other field-level attributes.**
Applies to commands:	**CVTDBFXLS, CVTDBFHTML, CVTDBFXML, CVTDBFCSV, CVTDBFTXT**
Dependent on:	**None**
Supports Slipstream variables	**No**

The FLDSTYLES parameter allows you to define, for individual fields in the input file, what style should be applied as well as a number of other field-level attributes.

The single value *DFT indicates that Slipstream will take all defaults. No field-level overrides will occur.

Field name and Record format name

Specifies the name of the field from the input file that the attributes defined here refer to.

Example:

```
CVTDBFXLS
FROMFILE(QADSPOBJ)
...
FLDSTYLES( (ODLBNM *FLDNAM library))
```

Here a style called “library” is being applied to the ODLBNM field.

Where a field name is entered here, the record format name defaults to *ONLY. If more than one record format has been selected on the RCDFMT parameter, *ONLY is invalid and you will need to specify the record format in which the field can be found.

Note that when using a Query/400 query, QM query or an SQL statement as the source of the data to be converted, Slipstream outputs the data to a temporary “outfile”, then converts the data in the outfile. In the course of this process, field names may change from those in the files referenced by the query or SQL.

For example, the following piece of SQL queries the outfile from the DSPOBJD command to obtain a list of libraries and the total size of the objects in each library, in descending order of total size.

```
SELECT
    ODLBNM, SUM(ODOBSZ)
    FROM QADSPOBJ
    GROUP BY ODLBNM
    ORDER BY 2 DESC
```

When the results of this SQL statement are saved in an outfile, the field containing the total size of the objects, corresponding to SUM(ODOBSZ), is given the name **SEL1** by SQL. It is this field which would need to be specified on the FLDSTYLES parameter to apply styling to this field.

Example:

```
CVTDBFXLS
FROMFILE(*SQL)
...
SQL('select odlbnm,sum(odobsz) from qadspobj group by odlbnm order by 2 desc')
...
FLDSTYLES( (SEL1 *FLDNAM *DATA *HEADER 'Total of Object Sizes' 14))
```

Here, an SQL statement is being run against the DSPOBJD outfile to provide input to the conversion. The FLDSTYLES parameter is used to apply a column heading and a column width to the second column in the SQL statement, which SQL names SEL1.

In order to make it easier to use this parameter in these circumstances, it is possible to refer to the field to which the styling should apply by field number within the record. To do this, specify the special value ***FLDNBR** on this field name element and specify the field number on the "Field number" element below.

Field number

Where *FLDNBR was specified for field name element above, indicating that you wish to identify the field to which this parameter is referring by means of a field number rather than a field name, specify the field number on this element.

Note that field numbers refer to the relative position of the field in the record format, counting from 1 for the first field. While this will be the same as the column number in an SQL statement, it is not necessarily the same as the column number in the output file (Excel, HTML etc.), as columns may have been excluded using the EXCLFLD parameter, or their sequence changed on the INCLFLD parameter.

Options are:

***FLDNAM**	(Default). The field is identified by means of its field name, specified on the previous element, not a field number.
field_number	If *FLDNBR was specified for the field name on the previous element, specify the field number here.

Example:

```
CVTDBFXLS
FROMFILE(*SQL)
...
SQL('select odlbnm,sum(odobsz) from qadspobj group by odlbnm order by 2 desc')
...
FLDSTYLES( (*FLDNBR 2 *DATA *HEADER 'Total of Object Sizes' 14))
```

Here, an SQL statement is being run against the DSPOBJD outfile to provide input to the conversion. The FLDSTYLES parameter is used with the *FLDNBR option to apply a column heading and a column width of 14 characters to the second column in the SQL statement, namely SUM(ODOBSZ).

Element or attribute

(XML only)

Whether the field should be output as an XML element or attribute.

Options are:

***DFT**	(Default). The field will be converted to a sub-element within the row element or an attribute of the row element depending on the value of the **Cols as elements or attributes** option of the XML parameter.
***ELEMENT**	The field will be converted to a sub-element of the row element.
***ATTRIBUTE**	The field will be converted to an attribute of the row element.

Element or attribute name

(XML only)

The name of the sub-element or attribute this field is converted to.

Options are:

***DFT**	(Default). The name of the sub-element or attribute this field is converted to will be generated based on the value of the **Generate elem/attr names from** option of the XML parameter.
***FLDNAM**	The name is generated from the field name
***QUALFLD**	The name is generated from the qualified field name (field name and record name).
***ALIAS**	The name is generated from the field alias.
***QUALALIAS**	The name is generated from the qualified field alias (field alias and record name).
name	Specify the name to use

Data style name

(Excel, HTML and XML only)

The name of the style to be applied to this field in data rows. Use this option to apply a different named style to the data rows for individual columns.

Options are:

***DATA**	(Default). The predefined *DATA style is applied.
style_name	Specify the name of a predefined or user-defined style to be applied to his field in data rows.

Header style name

(Excel, HTML and XML only)

The name of the style to be applied to this field in header rows. Use this option to apply a different named style to the header rows for individual columns.

Options are:

***HEADER**	(Default). The predefined *HEADER style is applied.
style_name	Specify the name of a predefined or user-defined style to be applied to his field in header rows.

Column heading

The column heading text for this field.

Options are:

***HEADER**	(Default). The values defined on the HEADER parameter will dictate the heading text.
***AVAIL**	Slipstream will select the best available labels from the input file to create the header row. This selection of label text is performed according to the following criteria. If the fields in the input file have Column Headings (DDS COLHDG keyword), they will be used to generate the column headings. Otherwise, if the fields in the input file have field aliases (DDS ALIAS keyword), those will be used instead. Otherwise, if the fields in the input file have text descriptors (DDS TEXT keyword), those will be used. Otherwise the field names will be used. For fixed-length ASCII text files, no header row will be created.
***NONE**	No header row is created.
***COLHDG**	For CSV, Excel and HTML files, a header row is created from the field column headings (DDS COLHDG keyword). This option is invalid if TOFMT(*FIXED) is specified.
***ALIAS**	For CSV, Excel and HTML files, a header row is created from the field aliases (DDS ALIAS keyword). This option is invalid if TOFMT(*FIXED) is specified.
***TEXT**	For CSV, Excel and HTML files, a header row is created from the field text descriptors (DDS TEXT keyword). This option is invalid if TOFMT(*FIXED) is specified.
***FLDNAM**	For CSV, Excel and HTML files, a header row is created from the field names. This option is invalid if TOFMT(*FIXED) is specified.
***COLHDG1**	The first column heading element only.
***COLHDG2**	The second column heading element only.
***COLHDG3**	The thrid column heading element only.
***COLHDG12**	Column heading elements 1 and 2 only.
***COLHDG13**	Column heading elements 1 and 3 only.
***COLHDG23**	Column heading elements 2 and 3 only
heading_text	Specify the heading text required.

Column width

(Excel only. This attribute has not been implemented for HTML and XML because browser behavior and support in this area is just too variable and unreliable).

Set the width of the column for this field.

Options are:

***DFT**	(Default). Calculate the width of the column in the default manner. For Excel, this means using the method specified on the **Column width option** element of the EXCEL parameter.
***FIELDSIZE**	Base the column width on the size of the field, according to its DDS definition.

***AUTOFIT**	Base the column width on the width of the largest data value.
column_width	Specify the column width in characters. For Excel, this can be difficult for Slipstream to calculate as it is dependent on the metrics of the fonts that are being used, which may not be available .

Format of num/char date/time

Indicates that the field in question contains a date or time and specified the format of this date or time.

If you have a file that contains dates or times in character fields (for example the outfile created by the DSPOBJD command), or times held in numeric fields, the only way to tell Slipstream that these fields should be treated as dates or times is to specify those fields on this parameter and define their format on this element.

If you have a file that contains dates in numeric fields, and Slipstream does not recognize those fields as dates, (for example, because they have an unusual size or number of decimals or no date edit code/word), you can define those fields and their date formats here.

Similarly, if your file contains some numeric dates in one format (e.g. *MDY) and some in another (e.g. *YMD), you can specify the most commonly used date format on the DBFDATFMT parameter and list the fields that use a different format here.

Options are:

***NONE**	The field is not a date.
***YMD**	The field is a date in YYMMDD or YYYYMMDD format
***DMY**	The field is a date in DDMMYY or DDMMYYYY format
***MDY**	The field is a date in MMDDYY or MMDDYYYY format
***CYMD**	The field is a date in CYYMMDD format
***CDMY**	The field is a date in CDDMMYY format
***CMDY**	The field is a date in CMMDDYY format
***HMS**	The field is a time in HHMMSS format.

Example:

```
CVTDBFXLS FROMFILE(QADSPOBJ)
...
 FLDSTYLES( (ODCDAT *N *DATA *HEADER *HEADER *DFT *MDY)
            (ODSRCD *N *DATA *HEADER *HEADER *DFT *YMD)
            (ODCTIM *N *DATA *HEADER *HEADER *DFT *HMS))
```

Here we are converting the outfile from the DSPOBJD command. The creation date field ODCDAT is flagged to Slipstream as a date in *MDY format inside a character field. Similarly, ODSRCD (source change date) is also a date inside a character field, but is in *YMD format. Finally, ODCTIM (the creation time) is identified as a time value inside a character field.

Separator of char date/time

Where the field is a date or time inside a character field, this element defines the separator used to format the date or time.

Options are:

***NONE**	(Default). The character date does not contain a separator.
***SYSVAL**	The separator is the character defined by the QDATSEP system value
***JOB**	The separator is the character defined by the DATSEP job attribute
***SLASH**	/
***HYPHEN**	-
***PERIOD**	.
***COMMA**	,
***COLON**	:
***BLANK**	Space
sep_char	Specify the separator character

SORT – Sort specifications

Parameter	**SORT**
Description	**Defines how the data should be sorted**
Applies to commands:	**CVTDBFXLS, CVTDBFCSV, CVTDBFHTML, CVTDBFPDF, CVTDBFXML, CVTDBFTXT** **CVTDBFSTMF (deprecated)**
Dependent on:	**None**
Supports Slipstream variables	**No**

The **SORT** (Sort Specifications) parameter allows you change the order in which records retrieved from the input source are presented in the output stream file by sorting those records by one or more fields.

Up to 50 sort keys may be specified in the format explained below, or one of these two single values can be selected:

***FILE**	(Default). The file keys are used to sequence the records. If the file has no keys, arrival sequence is used.
***NONE**	Arrival sequence is used. The records will be presented in the sequence in which they appear physically in the file.

Each sort key has two elements:

Key field name

Field_name	Specify the name of the field to use as a sort key.

Key field order

***ASCEND**	(Default). Sort in ascending sequence.
***DESCEND**	Sort in descending sequence.

QRYSLT - Query selection expression

Parameter	**QRYSLT**
Description	**Defines selection criteria**
Applies to commands:	**CVTDBFXLS, CVTDBFCSV, CVTDBFHTML, CVTDBFPDF, CVTDBFXML, CVTDBFTXT** **CVTDBFSTMF (deprecated)**
Dependent on:	**None**
Supports Slipstream variables	**Yes**

The **QRYSLT** (Query Selection Expression) parameter allows you select and exclude records retrieved from the input source by applying criteria that you define here.

***ALL**	(Default). All records in the input source are selected.
'Query_selection'	Specify an expression of up to 512 characters (contained in apostrophes) that describes the values used to determine which records are selected. To expand the field beyond 512 characters to a maximum of 5000 characters, you must specify the parameter on the command entry display. You can specify any logical expression formed from relationships (such as *EQ and *NE) of field and constant values or functions of field and constant values. At least one field name is specified in each relationship.

The syntax of this query selection string is identical to that used by the OPNQRYF command QRYSLT parameter. See http://publib.boulder.ibm.com/iseries/v5r2/ic2924/index.htm?info/dbp/rbafomst207.htm for further details of how to use this parameter.

Please note that where the input to CVTDBFXLS is one of the special values *SQL, *SQLSRC, *QMQRY or QRYDFN, any record selections made in the SQL or query will reduce the size of the temporary results file which CVTDBFXLS will process as its input. The ***Slipstream*** QRYSLT parameter then allows you to apply record selection to the temporary results file if that is appropriate.

For example, you could use the SQL to query your order file and calculate order totals by customer by means of the GROUP BY function. You could use the SQL WHERE clauses to select only customers in New York State. The temporary results file would contain totals for each customer. You could then use the ***Slipstream*** QRYSLT parameter to select only those customers whose totals exceed a given threshold value.

DBFCCSID – Database file CCSID

Parameter	**DBFCCSID**
Description	**Defines the encoding of data in the input file**
Applies to commands:	**CVTDBFXLS, CVTDBFCSV, CVTDBFHTML, CVTDBFPDF, CVTDBFXML, CVTDBFTXT** **CVTDBFSTMF (deprecated)**
Dependent on:	**None**
Supports Slipstream variables	**No**

The DBFCCSID (Database File CCSID) parameter enables you to specify the CCSID (Coded Character Set Identifier) which best describes the encoding of data in the database file when this information is not otherwise available to Slipstream.

There are two elements to this parameter.

CCSID

The first element indicates the CCSID to be used where the encoding of the data cannot be determined from the metadata associated with the database file.

Options are:

***DBF**	The CCSID is determined from the information available in the database file metadata (DDS).
***JOB**	The CCSID of the current job is used.
***SYSVAL**	The value of the QCCSID system value is used.
***USER**	The CCSID attribute of the user profile of the user running the command is used.
CCSID_value	Specify a CCSID to use.

Override file CCSID?

The second element indicates whether this CCSID should be used only where no CCSID information is available, or whether it should be used in all cases and override any CCSID information available from the metadata (DDS) associated with the database file.

Options are:

***NO**	Use the CCSID only where no CCSID information is available in the database file metadata (DDS).
***YES**	Use the CCSID for all data, disregarding any CCSID information available from the metadata. This option can be useful where the encoding of the file is different from that suggested by the CCSID information associated with it.

STMFCODPAG – Stream file code page

Parameter	**STMFCODPAG**
Description	**Defines the encoding of data in the output file**
Applies to commands:	**CVTDBFCSV, CVTDBFHTML, CVTDBFXML, CVTDBFTXT** **CVTDBFSTMF (deprecated)**
Dependent on:	**None**
Supports Slipstream variables	**No**

The STMFCODPAG (Stream File Code Page) parameter controls the way system i EBCDIC character data is converted to ASCII for inclusion in the stream file.

Options are:

***CALC**	Slipstream calculates the most appropriate code page for the stream file based on the CCSID of the data in the database file and the format to which the data is being converted.
***PCASCII**	EBCDIC data is converted to a Windows ASCII codepage corresponding to the EBCDIC CCSID of the data in the input file.
***WINDOWS**	The same as *PCASCII.
***STDASCII**	EBCDIC data is converted to an ASCII codepage corresponding to the EBCDIC CCSID of the data in the input file.
***IBMASCII**	The same as *STDASCII.
***ISOASCII**	EBCDIC data is converted to an ISO ASCII codepage corresponding to the EBCDIC CCSID of the data in the input file.
UNICODE	EBCDIC data is converted to a Unicode codepage corresponding to the EBCDIC CCSID of the data in the input file. UCS2 will be used.
***UCS2**	Universal Character Set encoded in 2 octets. (a form of Unicode that consistently uses 2 bytes to represent characters).
***UTF8**	UCS Transformation Format 8, a form of Unicode which uses 1, 2 or 4 bytes to represent characters.
***UTF16**	UCS Transformation Format 16.
***NOCONV**	Data is not converted and is left in its original format. This may give the best results with some languages such as Arabic.
CCSID	Specify a CCSID between 1 and 65535 representing the encoding scheme to which data is converted.

Note that when converting to a format such as XML, Slipstream does not force you to select an encoding which is likely to be immediately usable. For example, there is nothing to stop you from specifying CVTDBFXML ... STMFCODPAG(37) to create an XML file encoded in

US EBCDIC, but it is unlikely that XML consumers will be able to process the file in that encoding.

UNICODE

Parameter	**UNICODE**
Description	**Unicode-related options**
Applies to commands:	**CVTDBFCSV, CVTDBFHTML, CVTDBFXML, CVTDBFTXT** **CVTDBFSTMF (deprecated)**
Dependent on:	**STMFCODPAG(*UNICODE)** **STMFCODPAG(*UTF8)** **STMFCODPAG(*UTF16)** **STMFCODPAG(*UCS2)**
Supports Slipstream variables	**No**

The UNICODE (Unicode Options) parameter allows you to define various Unicode-related options when you specify a Unicode encoding scheme on the STMFCODPAG parameter.

There are two elements to this parameter.

Bigendian or littleendian

In relation to UCS2 encoding, which uses two bytes to represent each character, this option determines the order in which the bytes are represented.

Options are:

***BIG**	Bigendian representation, with the most significant byte first (system i norm).
***LITTLE**	Littleendian representation, with the least significant byte first (PC norm)

Include Unicode marker?

This option determines whether Slipstream outputs a marker at the start of a text file which indicates to a reader application whether the byte order is bigendian or littleendian.

Some applications such as Windows NotePad check for a marker at the start of the file (hex x'FEFF', x'FFFE') and use this to identify whether UCS2 or UTF-16 Unicode data is encoded in bigendian or littleendian format. The marker x'EFBBB' denotes UTF-8 encoding.

Options are:

***TOFMT**	(CVTDBFSTMF only) A marker is output if the TOFMT parameter if *FIXED or *HTML, but not for *CSV.
***YES**	A marker is output
***NO**	No marker is output

DBFDATFMT – Database date format

Parameter	**DBFDATFMT**
Description	**Specifies the format in which dates are held in numeric fields in the input data**
Applies to commands:	**CVTDBFCSV, CVTDBFHTML, CVTDBFXML, CVTDBFTXT, CVTDBFPDF, CVTDBFXLS** **CVTDBFSTMF (deprecated)**
Dependent on:	**None**
Supports Slipstream variables	**No**

The DBFDATFMT (Database Date Format) parameter allows you to define the format in which dates are held in numeric fields in the database file you are converting.

If a field in the database file being converted is:

- Packed decimal or zoned decimal
- 6, 7 or 8 digits in length
- zero decimal places
- edited with an edit code of 'Y'

Slipstream will attempt to interpret the field as a date and convert it to a date in the output file. In doing this, Slipstream needs to know the date format in which the data is held in the field.

Note that if you have dates inside character fields (e.g. field ODCDAT in DSPOBJD outfile QADSPOBJ) or if you have some dates in numeric fields which are in one format and some in another (e.g. some are held *MDY and some *YMD), you will need to specify the format of those dates individually on the FLDSTYLES parameter. This parameter defines the default assumed format for dates in numeric fields only.

Options are:

***ENVVAR**	Slipstream will use the date format defined in the environment variable SL_DBFDATFMT. If that environment variable exists, and is set to one of the values below, that value is used. If the environment variable does not exist, or if it contains anything other than one of the values listed below, *YMD is assumed.
***YMD**	Slipstream will assume the date is in YYMMDD format if the field is 6-7 digits long, and in YYYYMMDD format if the field is 8 digits long.
***NONE**	Slipstream will not attempt to convert numeric fields that appear to be dates to date. They will be converted as numbers.
***DMY**	Slipstream will assume the date is in DDMMYY format if the field is 6-7 digits long, and in DDMMYYYY format if the field is 8 digits long.

***MDY**	Slipstream will assume the date is in MMDDYY format if the field is 6-7 digits long, and in MMDDYYYY format if the field is 8 digits long.
***CYMD**	CYYMMDD format, where C is 0 for the 20th century and 1 for the 21st
***CDMY**	CDDMMYY format, where C is 0 for the 20th century and 1 for the 21st
***CMDY**	CMMDDYY format, where C is 0 for the 20th century and 1 for the 21st

AUT

Parameter	**AUT**
Description	**Indicates the public authority to be given to the new file.**
Applies to commands:	**CVTDBFCSV, CVTDBFHTML, CVTDBFXML, CVTDBFTXT, CVTDBFPDF, CVTDBFXLS** **CVTDBFSTMF (deprecated)**
Dependent on:	**None**
Supports Slipstream variables	**No**

The **AUT** (Public Authority) parameter allows you to control the public authority given to the stream file when it is created.

Options are:

***R**	(Default). Read only
***W**	Write only
***X**	Execute only
***RW**	Read and write
***RX**	Read and execute
***WX**	Write and execute
***RWX**	Read, write and execute (all)
***NONE**	No authority
autl_name	Specify the name of an authorization list that will control public authority to the file

Note that in previous releases Slipstream set the owner's authority to *RWX by default and set the public authority based on this parameter, but did not set the group authority. This has been corrected in V5R2M0 and the group authority is set to *RWX in all cases.

INHERITAUT

Parameter	**INHERITAUT**
Description	**Indicates whether authorities are inherited from the parent directory in which the stream file is created.**
Applies to commands:	**CVTDBFSTMF (deprecated)**
Dependent on:	**None**
Supports Slipstream variables	**No**

The INHERITAUT (Inherit Authority) parameter allows you to control whether object authorities are inherited from the parent directory in which the stream file is created.

This parameter exists only on CVTDBFSTMF. The format-specific commands always inherit authorities from the parent directory (equivalent to CVTDBFSTMF ... INHERITAUT(*YES)).

Options are:

__*NO__	Authorities are not inherited from the directory
***YES**	Authorities are inherited from the directory

When you specify INHERITAUT(*NO), the object authorities (*OBJEXIST, *OBJMGT, *OBJALTER, and *OBJREF) assigned to the owner, primary group, and *PUBLIC in respect of the stream file being created are copied from the owner, primary group, and public object authorities of the parent directory in which the stream file is created. This occurs even when the new file has a different owner than the parent directory. The new file does not have any private authorities or authorization list. It only has authorities for the owner, primary group, and public. The owner is assigned full data authorities and *PUBLIC is assigned the data authorities specified on the AUT parameter.

When you specify INHERITAUT(*YES), the object authorities (*OBJEXIST, *OBJMGT, *OBJALTER, and *OBJREF) assigned to the owner, primary group, and *PUBLIC in respect of the stream file being created are copied from the owner, primary group, and public object authorities of the parent directory in which the stream file is created. However, the private authorities (if any) and authorization list (if any) are also copied from the parent directory. If the new file has a different owner than the parent directory and the new file's owner has a private authority in the parent directory, that private authority is not copied from the parent directory. The authority for the owner of the new file is copied from the owner of the parent directory. The owner is assigned full data authorities and *PUBLIC is assigned the data authorities specified on the AUT parameter.

CVTXLSDBF Command

The CVTXLSDBF command lets you extract the contents of cells in an Excel spreadsheet. The data is written in a standard database “outfile” format. Your applications can easily read this outfile and process its contents, for example to import the data into your own database.

This provides a convenient means of getting data stored in an Excel spreadsheet into an AS/400 database without the need for a PC. Thus, processing can be carried out automatically, and without user intervention, perhaps as part of your overnight batch job suite. For example, users might enter information into a spreadsheet during the day which is subsequently picked up and processed by your program after close of business. Your program might use this command to extract the data from the spreadsheet and write it to your AS/400 database files or perform other processing on it.

Note that the outfile contains null-capable fields and ILE RPG programs will need to have the header specification (H-spec) **ALWNULL(*INPUTONLY)** or **ALWNULL(*USRCTL)** defined, or specify the equivalent value on your compile command, in order to process the file.

See below for a discussion of the format in which the data extracted from your spreadsheet is stored.

The command parameters are as follows.

FROMSTMF – Excel file to convert

Specifies the path name of the Excel file from which the cell contents will be extracted.

FROMSHEETS –Worksheet(s) to convert

Specifies one or more worksheets in the Excel spreadsheet that from which cell contents will be extracted.

The default is the special value:

***ALL**	Data will be extracted from all worksheets in the workbook.

TOFILE – File to receive output

The name of the database file which will receive the output.

If the file does not exist, it will be created.

If the file does exist, it must be a file in the correct format (for example, a file previously created by this command). If the file is not in the correct format, an error will occur.

The default is the special value:

***FROMSTMF**	The name of the file will be derived from the name of the spreadsheet specified on the FROMSTMF parameter by removing any extension from the file name part of the path name and remainder of the file name up to a maximum of 10 characters.

Options for the library name are:

***LIBL**	Slipstream will use the library list to locate an existing file. If no file is found, a new file will be created in the current library.

***CURLIB**	Slipstream will look for an existing file in the current library, or create a new file in the current library.
library_name	Specify the library in which an existing fie is located or in which to create a new file.

TOMBR – Member to receive output

Name

The name of the member in the database file which will receive the output.

If the member does not exist, it will be added to the file.

If the file does exist, data in that member will be replaced or the new data will be appended to that member depending on the setting of the “Replace or add records” option below..

Options are:

***FROMSTMF**	The name of the member will be derived from the name of the spreadsheet specified on the FROMSTMF parameter by removing any extension from the file name part of the path name and remainder of the file name up to a maximum of 10 characters.
***FIRST**	The data will be written to the first member in the file. If there are members in the file, a new member with the same name the file will be added and used.
member_name	Specify the name of the member to receive the data.

Replace or add records

Determines whether any existing data in the file is replaced or whether the new data is appended to any existing data.

Options are:

***REPLACE**	Any existing data in the file is replaced.
***ADD**	The new data is appended to any existing data.

TOCCSID – CCSID to convert to

Specifies the CCSID (character encoding) to which the cell contents will be converted. Data in Excel files is typically stored in ASCII or unicode and you will normally wish to convert that data to an appropriate EBCDIC CCSID for processing on the system i.

Options are:

***JOB**	The CCSID of the current job.
***SYSVAL**	The CCSID indicated by the QCCSID system value.
***USER**	The CCSID associated with the current user’s user profile.
CCSID	Specify a CCSID to use.

BLANKS – Output blank cells

Determines whether anything is written to the output for cells that contain blanks (Excel BLANK and/or MULBLANK records). These records typically denote empty cells that

contain no data and are not the same as cells that contain labels that consist of all blank (space) characters).

Options are:

***NO**	Cells that are defined but empty (represented by contain Excel BLANK or MULBLANK records) are ignored and not written to the output.
***YES**	Cells that are defined but empty (represented by contain Excel BLANK or MULBLANK records) are not ignored and written to the output.

ROUNDING – Floating point rounding

Specifies the number of decimal places to which floating point values are rounded.

Numeric values in cells in an Excel spreadsheet are typically held either as integer values or as floating point values. These are written to the output file in packed decimal format as that format is usually more convenient for system i applications to process than floating point. However, the process of converting floating point numbers to packed decimal can result in a very small loss of fidelity, which is no. In order to avoid

Options are:

5	Floating point values are rounded to 5 decimal places before being written to the file.
decimal_places	Specify between 0 and 9 decimal places to which floating point values will be rounded.

CVTXLSDBF output format

The format of files created by CVTDBFXLS are described below.

These files are organized in such a way that data from any Excel spreadsheet can be written to them and processed by application programs, irrespective of the layout of the original spreadsheet. Inevitably, to achieve this level of flexibility, the design of this generic outfile has involved a degree of compromise. The structure is denormalized in order to allow all data to be written to a single file and data is consequently not held as efficiently as it might be. You should be aware, therefore, that the outfile created when extracting information from a large spreadsheet could occupy a significant amount of storage.

The basic organization of the file is that each record represents a single cell in the spreadsheet, but only cells that contain data are represented in the outfile (whether cells which contain empty values are represented is controlled by the BLANKS parameter – see above).

The file that is created is a multi-member uniquely keyed physical with a record format called SL_XLSOUTR. Each record in the file is uniquely identified by a combination of sheet number, row number and column number.

The layout of records in the file is as follows:

Field name	Format	DDS definition	Description	Comments
SHEETNBR	BINARY(4)	9B 0	Sheet number	Identifies the worksheet in the Excel workbook from

				which the cell data was extracted. Worksheets are numbered from 1 starting with the first worksheet in the workbook.
SHEETNAME	CHAR(32)	32A	Sheet name	Identifies the worksheet in the Excel workbook from which the cell data was extracted. The worksheet name repeated in each record for reasons of convenience in order to make it possible to write all data to a single outfile.
ROWNBR	BINARY(4)	9B 0	Row number	Identifies the row in the worksheet from which the cell data was extracted. Note that rows are numbered from 1 as per the Excel user interface.
COLUMNNBR	BINARY(4)	9 B 0	Column number	Identifies the column in the worksheet from which the cell data was extracted. Note that columns are numbered from 1, equivalent to column A in the Excel user interface. Columns numbers rather than letters are used here to denote cells in the worksheet in order to ensure proper sequencing. For example, column Z must precede column AA. In this file, column Z is numbered 26 and column AA is numbered 27 in order to sort records correctly.
COLUMNREF	CHAR(2)	2A	Column reference	The column reference of the column identified by COLUMNNBR above, in the form in which it is usually represented, i.e. a letter A-Z or pair of letters AA-ZZ.
DATATYPE	CHAR(1)	1A	Data type	The data type of the cell value. The value of this field determines which (if any) of the fields NUMBER and TEXT contains the actual cell value. Options are: N = Numeric. The cell contains a numeric value and that value is stored in the field NUMBER below. A = Alphanumeric. The cell contains a text label and that

				label value is stored in the field TEXT below. B = Blank. The cell is defined but empty. Neither NUMBER not TEXT will contain a value.
NUMBER	PACKED(31,9)	31P 9	Numeric value	If the Excel cell contains a floating point or integer value, that numeric value is returned in this field, and DATATYPE contains an “N”, otherwise this field is NULL.
TEXTCCSID	BINARY(4)	9B 0	CCSID of text value	If the Excel cell contains a text label, this field holds the CCSID to which that text value was converted before being stored in TEXT below, otherwise this field is NULL.
TEXT	VARLEN(1024)	1024A (varying)	Text value	If the Excel cell contains a text label, that text value is returned in this field, and DATATYPE contains an “A”, otherwise this field is NULL.

Example:

Imagine a spreadsheet called **example.xls** containing a single worksheet called Sheet1 with the following contents:

	A	B
1	Column A text	123.456
2	123	Column B text
3	456	789.012

Running the following command:

CVTXLSDBF FROMSTMF(example.xls)

Creates a file called EXAMPLE in the current library which contains the following data:

Sheet Number	Sheet Name	Row Number	Column Number	Column Ref	Data Type	Numeric Value	CSSID of Text Data	Text Value
1	Sheet1	1	1	A	A		37	Column A text
1	Sheet1	1	2	B	N	123.456		
1	Sheet1	2	1	A	N	123		
1	Sheet1	2	2	B	A		37	Column B text
1	Sheet1	3	1	A	N	456		
1	Sheet1	3	2	B	N	789.012		

The value of the TEXTCCSID field will be dependent on the job CCSID attribute.

DSPENCPWD Command

The DSPENCPWD command displays a string of hexadecimal digits representing the encrypted form of a password that you enter.

You can use this encrypted form of the password on a number of parameters of the CVTDBFxxx commands (CVTDBFXLS, CVTDBFXML etc.). Slipstream will decrypt the password before using it. This provides a means of avoiding the need to store passwords in source code in plain text form.

The following parameters support the use of encrypted passwords:

Command	Parameter	Element	Description
CVTDBFXLS CVTDBFXML CVTDBFTXT CVTDBFHTML CVTDBFPDF CVTDBFCSV	FTP	Remote password	Password for the FTP server
CVTDBFXLS CVTDBFXML CVTDBFTXT CVTDBFHTML CVTDBFPDF CVTDBFCSV	EMAILOPT	Zip file password	Optional password of the zip file in which email attachments may be sent
CVTDBFXLS	XLSPROTECT	Worksheet unprotect password	Password needed to unprotect a protected worksheet
ZIPDTA	PWD	Password	Password for zip file

In each case, the parameter element is followed by another element called **"Encrypted password supplied"**. If *YES is specified for this element, Slipstream will interpret the password supplied on the previous element as an encrypted password string and wil automatically decrypt it using the internal key setting before using the password.

The password algorithm used is not symmetrical, i.e. simply supplying the encrypted form of the password to DSPENCPWD will not result in the original password being displayed.

The command parameters are as follows.

PWD– Password to encrypt

Specify the password to be encrypted. Slipstream will apply an encryption algorithm to the password string using an internal key setting and will display the password in its encrypted form as a string of hexadecimal digits. Make or note of this encrypted password string for use in your code or copy and paste it into your source member.

ZIPDTA Command

The ZIPDTA command compresses one or more files (IFS stream files or database file members) using the industry-standard ZLIB compression library to create a zip file compatible with applications such as WinZip.

zlib was written by Jean-loup Gailly (compression) and Mark Adler (decompression).

Command parameters are as follows:

FROMFILE –Files to zip

Specify from one to 100 file path names in IFS naming format.

Path names can be generic, .e.g.

/home/myfiles/*

or /home/myfiles/test*

or /home/myfiles/*.dat

or /home/myfiles/test*.d*

Physical file members are supported but must be specified in IFS naming format. For example, to zip all members in file MYFILE in library MYLIB, use:

ZIPDTA FROMFILE('/qsys.lib/mylib.lib/myfile.file/*')

Note that ZIPDTA simply compresses the data in a physical file member: it does not carry out any conversion or re-encoding. This means that if you zip a physical file member on the system i (most probably EBCDIC) and attempt to unzip on an ASCII system (e.g. PC or UNIX machine) the data will unzip but is unlikely to be easily readable.

TOZIP – Zip file name

The name of the zip file which be created or replaced.

STMFOPT – Stream file option

Whether an existing file will be replaced or the new zipped data added to an existing zip file.

Options are:

***NONE**	If the file specified on the TOZIP parameter already exists, it will not be replaced and no changes will be made to it.
***REPLACE**	If the file specified on the TOZIP parameter already exists, it will be replaced.
***ADD**	If the file specified on the TOZIP parameter already exists, the newly compressed files will be added to it.

CPRLVL – The data compression level

The level of data compression that is applied. The higher the compression level, the smaller the files that result (normally) but the longer the compression processing time.

Options are:

***DFT**	The default compression level (6)
***FASTEST**	The fastest compression method but giving the least compressed results (1).

***BEST**	The method giving the highest compression factor but also taking the longest to process (9)
***NONE**	No compression (0). Files are stored in the zip uncompressed.
compression	Specify the compression level (0-9)

PWD – Password

An optional password for the zip file. If a password is specified, the zipped file will be encrypted.

There is a single option:

***NONE**	No password is required to unzip the file and the file is not encrypted.

Password

The password needed to unzip the file.

Encrypted password supplied

Whether or not the password supplied on the previous element is supplied in the encrypted form returned by Slipstream's DSPENCPWD (Display Encrypted Password) command.

DSPENCPWD applies an encryption algorithm to a password and returns a scrambled version of that password to you. If you specify the scrambled password on the previous element, and specify *YES here, Slipstream will unscramble the password for you before sending it to the FTP server. The main purpose of this facility is to avoid the need to hold passwords in plain text form in source code.

Options are:

***NO**	The password supplied on the previous element is in plain text format and not scrambled.
***YES**	The password supplied on the previous is in the scrambled form returned by DSPENCPWD. It will be automatically unscrambled before being sent to the FTP server.

UNZIPDTA Command

The UNZIPDTA command decompresses one or more files zipped using the industry-standard zip format. It uses the ZLIB compression library written originally by Jean-loup Gailly (compression) and Mark Adler (decompression).

Command parameters are as follows:

FROMZIP –From zip file

Specify the name of the zip file containing the file(s) to unzip.

FROMFILE –Files to unzip

Specify from one to 100 file path names in IFS naming format.

Path names can be generic, .e.g.

/home/myfiles/*

or /home/myfiles/test*

or /home/myfiles/*.dat

or /home/myfiles/test*.d*

The path name specified must match the directory path inside the zip file.

TODIR – Unzip to directory

The name of the directory into which the files are unzipped.

Options are:

***FROMFILE**	The file will be restored to the same directory as that in which it is stored inside the zip.
dir_name	Specify the directory into which the file will be unzipped.

Note that a physical file name can be specified here if the zipped file being restored is a physical file member.

TOFILE – Unzip to file

The name of the file after it has been unzipped.

Options are:

***FROMFILE**	The name of the file will be the same as in the zip.
file_name	Specify the new name of the file after unzipping.

REPLACE – Replace existing files

Whether an existing file will be replaced or not.

Options are:

***NO**	Existing files are not replaced and an error will occur if a file of the same name already exists.
***YES**	Any existing file of the same name will be replaced.

PWD – Password

The password required to unzip the file.

There is a single option:

***NONE**	No password is required to unzip the file.

Password

The password needed to unzip the file.

Encrypted password supplied

Whether or not the password supplied on the previous element is supplied in the encrypted form returned by Slipstream's DSPENCPWD (Display Encrypted Password) command.

DSPENCPWD applies an encryption algorithm to a password and returns a scrambled version of that password to you. If you specify the scrambled password on the previous element, and specify *YES here, Slipstream will unscramble the password for you before sending it to the FTP server. The main purpose of this facility is to avoid the need to hold passwords in plain text form in source code.

Options are:

***NO**	The password supplied on the previous element is in plain text format and not scrambled.
***YES**	The password supplied on the previous is in the scrambled form returned by DSPENCPWD. It will be automatically unscrambled before being sent to the FTP server.

STMFCODPAG – Stream file code page

The code page that will be allocated to a new file that is unzipped.

Note that this attribute does not in any way cause conversion or re-encoding of the contents of the file. It simply determines the setting of the CCSID attribute of the new stream file. You should choose an appropriate setting that accurately reflects the encoding of the contents of the file.

If the file already exists, its CCSID attribute is unchanged and this parameter is ignored.

Options are:

***PCASCII**	A Windows code page is derived from the job CCSID and assigned to the file.
***STDASCII**	A standard ASCII code page is derived from the job CCSID and assigned to the file.
***ISOASCII**	An ISO ASCII code page is derived from the job CCSID and assigned to the file.
ccsid	Specify the CCSID to assign to the file.

AUT – Public data authority

The public data authority level to assign to a new file.

Options are:

***R**	(Default). Read only
***W**	Write only
***X**	Execute only

***RW**	Read and write
***RX**	Read and execute
***WX**	Write and execute
***RWX**	Read, write and execute (all)
***NONE**	No authority
autl_name	Specify the name of an authorization list that will control public authority to the file

The Integrated File System

The system i Integrated File System (IFS) provides a coherent, coordinated set of file systems which can used for storing a variety of data physically on the system i or for communicating with file systems on other platforms.

These files systems include the following that may be of use to you for storing stream files created by ***Slipstream*** locally on your system i or remotely on another computer (PC, UNIX server etc.)

"root"	The **"root" (/)** file system. This file system takes full advantage of the stream file support and hierarchical directory structure of the integrated file system. The root file system has the characteristics of the Disk Operating System (DOS) and OS/2 file systems. You should typically use the root file system if you want to store stream files created by ***Slipstream*** locally on the same system i where you run the CVTDBFXLS command.
QNTC	Windows NT Server file system. This file system provides access to data and objects that are stored on a PC running Windows NT 4.0 or higher. It allows system i server applications to use the same data as Windows NT clients. If you prefer not to store stream file locally on your system i, you can use QNTC to enable ***Slipstream*** to write stream files directly to a PC running NT, Windows 2000, Windows XP etc. instead. This can be an effective way of sharing data created by ***Slipstream*** amongst your users and customers.
NFS	Network File System. This file system provides you with access to data and objects that are stored on a remote NFS server. An NFS server can export a network file system that NFS clients will then mount dynamically. This may be an option for sending stream files to a UNIX server.
QNetWare	The QNetWare file system. This file system provides access to local or remote data and objects that are stored on a server that runs Novell NetWare 4.10 or 4.11 or to standalone PC Servers running Novell Netware 4.12, 4.10 4.11 or 5.0. You can dynamically mount NetWare file systems over existing local file systems.
QOpenSys	The open systems file system. This file system is compatible with UNIX-based open system standards, such as POSIX and XPG. Like the root file system, this file system takes advantage of the stream file and directory support that is provided by the integrated file

system. In addition, it supports case-sensitive object names.

QDLS — The document library services file system (previously know as “shared folders”). This file system provides access to documents and folders. Use this file system only if you have applications which require it. QDLS is significantly slower and has major limitations (e.g. in relation to naming) compared with the root file system.

QFileSvr.400 — This file system provides access to other file systems that reside on remote system i servers. You can use QFileSvr.400 to save output from ***Slipstream*** directly to another system i.

See http://publib.boulder.ibm.com/system i/V5R2/ic2924/info/rzaia/rzaiacon.htm for full details of the Integrated File System.

We will now focus a little more closely on the QNTC File System since this is little known area of system i functionality which may well be of serious interest to users of ***Slipstream***.

QNTC

The QNTC file system is a subdivision of the IFS (Integrated File System) that enables the system i to access file and device shares (e.g. printers and CDROM) drives on a remote NT system. Please note that contrary to a commonly held fallacy this is ***NOT*** restricted to the Integrated XSeries Server (aka IPCS or FSIOP).

Using the QNTC file system, your system i can read and write files that reside physically on a PC running Windows NT 4 or above. This means that ***Slipstream*** can output stream files directly to an NT server if you would prefer to store them there rather than in the root file system of your IFS.

To use QNTC, the only software you need other than the base operating system is TCP/IP Connectivity Utilities for system i 400 *(S722-TCI).* However, setting up QNTC can be tricky. For full setup information, refer to the article in the IBM Software Knowledgebase at http://www-912.ibm.com/s_dir/slkbase.nsf/1ac66549a21402188625680b0002037e/aea450153eebf8ff8625670f0072550f?OpenDocument&Highlight=0,QNTC

However, here is a quick overview of the steps you need to follow to set up QNTC. This example relates to V5R2.

Domain Names

First you must ensure that the domain name defined by your system i NetServer configuration matches your PC’s Windows network workgroup name. NetServer is the function on the system i that provides support for the Windows Network Neighborhood. You can use Operations Navigator (OpsNav) to set up and manage NetServer.

Please note that changing the system i domain name in NetServer may affect which PCs will be able to see the system i in their Network Neighborhood.

From the main OpsNav window, click the name of your machine, select File Systems, and then right-click File Shares. Choose Open system i NetServer from the menu to display the NetServer window. Right-click system i NetServer and choose Properties to display the Properties window, where you can change the domain. Click the General tab and press the Next Start button. A window appears where you can set properties that will be used the next time NetServer is restarted. In the domain name field, enter a name that matches the workgroup of the NT PC that you wish to access.

Now, if you are sure that NetServer is not currently in use, end and restart NetServer by clicking the Stop icon followed (once NetServer has fully ended) by the green triangle icon.

If you wish to change the Network Id on the PC side, this can be done through the Windows Control Panel. On Windows 2000, select Start, Settings, Control Panel, and System. Then choose the Network Identification tab, click the Properties button, select Workgroup, and set the workgroup to match the domain that NetServer on the system i was configured to use.

On XP, you'll find the option on the Computer Name tab.

On Windows NT, the option to change the workgroup name can be found if you select Start, Settings, Control Panel, and Network.

Note that you may have to reboot the PC for these changes to take effect.

User Id and Password

One other thing which must match between the system i and the NT PC is the user id and password you're going to use. It is vital that your system i user id be recognised by the NT PC as a valid network logon id, and that the passwords are the same on both the system i and the NT machine. It may well be advisable to create a special user id on the two platforms specifically for the purpose of communicating between them using QNTC. You can then ensure that when the password needs to be changed, it is changed on both systems, if this needs to be done manually.

Testing the Connection

To test your connection, first ensure that you are logged on to the system i using a shared user id/password (see above), then run the command WRKLNK '/QNTC/*' on the system i. This could take several minutes to complete the first time it is run. Your NT system should appear on the list. If it doesn't, you may be able to manually establish a connection to the NT system using command CRTDIR /Q*NT*C/ <*servername*>', substituting the name of your NT system for *servername.*

Find your NT system in this list and choose Display (option 5) in WRKLNK (Work with Object Links) to display your file shares on NT. If the file shares don't appear, double-check that your user IDs and passwords match exactly on both systems.

You access the QNTC file system by including QNTC and the name of your PC and share name in the path name you specify on TOSTMF (To Stream File) parameter of the CVTDBFXLS command.

For example, let's imagine you have a company server running Windows NT 4 or above and you have decided that this is a convenient place to save the output from ***Slipstream*** so that all of your users can have shared access to the data. Let's imagine that this server is called **NTServer** and that it has a file share name set up called **NTFiles**. Below the share name, there is a directory called **CustData**. If you wanted to convert data from your customer file and save it as an Excel spreadsheet in this location under the file name **customer_file.xls**, you would run a command something like this:

```
CVTDBFXLS    FROMFILE(custfile)
             TOSTMF('/QNTC/NTServer/NTFiles/CustData/customer_file.xls')
             TOFMT(*XLS)
```

Note that the path name starts with **/QNTC**. This indicates to the system i that you are referring to the QNTC File System. Following /QNTC is the name of the PC to which the data is to be sent: **NTServer**. Next comes the name of the file share: **NTFiles**. After that is the name of the directory below the share: **CustData**. Finally we have the name of the file itself: **customer_file.xls**.